EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE LAUSANNE
POLITECNICO FEDERALE DI LOSANNA
SWISS FEDERAL INSTITUTE OF TECHNOLOGY LAUSANNE

FACULTÉ DE SCIENCES DE BASE
SECTION DE MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Analysis and Optimization of Perfectly Matched Layers for the Boltzmann Equation

MASTER PROJECT (Spring 2015)

Author:
Marco SUTTI

Under the supervision of:
Professor Jan S. HESTHAVEN

Submitted on 19 June 2015

Abstract

Numerical simulation of problems defined on unbounded domains are challenging due to physical constraints on computational resources. When approaching this type of problem, one is forced to somehow truncate the simulation domain. In order to ensure consistency between the equation to solve and the computational model on a truncated domain, some form of absorbing boundary or boundary layer is often required.

In this work we study stability and optimization of an absorbing layer for the Boltzmann equation. More precisely, we look at the BGK approximation to the Boltzmann equation and study an absorbing layer developed following the perfectly matched layer (PML) technique.

First, a review of the theory behind the PML technique is presented along with a PML for the BGK model. The review is followed by an analysis on the stability of the model. We find that in order to ensure stability, some of the parameters in the model must be discarded. By employing the ANOVA expansion of multivariate functions we calculate the Total Sensitivity Indices of the remaining parameters of the model. Finally, a small set of important parameters is found and minimization techniques are used to choose the optimal parameter values in this set.

Keywords BGK model, perfectly matched layer, modal analysis, differential operators, stability analysis, ANOVA expansion, total sensitivity index

Acknowledgements

I owe my deepest gratitude to Professor Jan S. Hesthaven, supervisor of this project, for his constant assistance, suggestions and for giving me intellectual freedom in my work. I also express my gratitude to my friend and colleague Caterina who spent with me this period in the MCSS group and with whom I shared ideas and coffee breaks. My acknowledgements go also to the entire MCSS group, and in particular to Khemraj for some insightful discussions.

Special thanks go to my dear friend and colleague Matteo for the interesting discussions about mathematics. Thanks also to my friend Alessandro that did not show up at lunch meetings and made me walk to a completely wrong restaurant. It seems that pure mathematicians do really have their heads up in the clouds. I thank also my friend Irene for the nice chats and the delicious cakes, all the other friends of the CSE Master and my friends in Italy.

And finally, I thank my family for supporting me during my staying in Lausanne. My thanks go in particular to my parents Dario and Alda, my sisters Mara and Sonia, and my nieces Elisa, Giulia, Matilde, Emma and the newborn Maria Letizia. My sincerest gratitude go to all of them.

Marco Sutti

Contents

List of Figures

List of Tables

Introduction

Due to physical constraints on computational resources, numerical simulations of unbounded physical problems are virtually impossible to carry out without the truncation of the simulation domain. When not dealing with periodic solutions, one is usually forced to truncate the domain by introducing a boundary layer or absorbing layer.

In this work we study and enhance an effective absorbing layer for the Boltzmann equation. In particular, we will look at the BGK approximation to the Boltzmann equation and we will make use of the perfectly matched layers technique.

The concept of perfectly matched layer has been introduced by Bérenger [3] starting from physical considerations on electromagnetic waves. Bérenger changed Maxwell equations in the absorbing layer so that waves entering into the layer are damped out and no reflections arise at the interface, as Figure 1 illustrates. This is the reason why the layer is referred to as being perfectly matched.

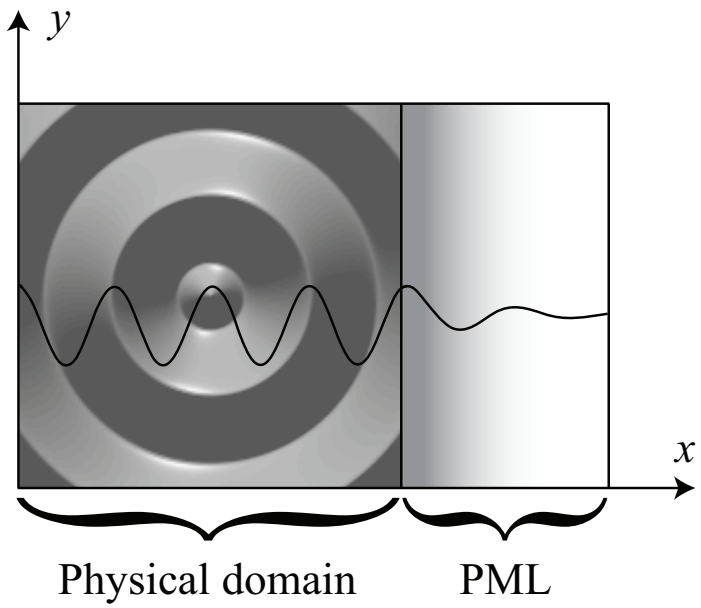


Figure 1: Qualitative illustration of a wave problem with an absorbing PML.

However, the original approach of Bérenger was based on a splitting technique that could break the hyperbolicity of the system. Then, if the problem is no more hyperbolic, but only *weakly* hyperbolic, the lower order terms must be treated carefully, because some disturbances may arise at later stages of the simulation.

A new construction of PMLs for hyperbolic systems was brought forth by Hagstrom [17]. He proposed a procedure based on the modal analysis of the governing equations in Laplace-Fourier space in order to derive the layer model, which yet is feasible only for linear low-order terms. According to this approach, the modal solution inside the layer is constructed so that the eigenfunctions of the problem remain the same regardless whether we are looking at the problem outside the layer or inside the layer. This guarantees that no reflection will arise at the interface and the layer is perfectly matched. Appelö et al. [2] later deepened the analysis of this technique and established a solid theory behind it. By using such approach, Gao et al. [11] constructed a PML for the BGK equations, and this is the model that we are going to use in the present work.

The purpose of this work is to carry out an analysis of the PML for the BGK equations proposed by Gao et al. [11] to investigate the role and importance of the parameters appearing in the model. To this aim we will use both analytical and numerical tools. The ultimate goal would be to be able to use the absorbing layer so developed together with the Navier-Stokes equations (NSE). There are actually some formulas that allow to relate the variables evolved by the BGK equations to the physical variables described by the NSE. However, the key point here is that although the NSE have a nonlinear nature, the BGK equations are *linear*. Since there is nowadays a well established theory for the development and the analysis of perfectly matched layers for *linear problems* [2, 17], it should now be apparent why we resort to the study of the BGK model. The coupling of the NSE and the BGK equations is left for future work.

Outline

The rest of this work is organized as follows. Chapter 1 presents the Bhatnagar-Gross-Krook (BGK) model of the Boltzmann equation, and details the implementation aspects and the tests performed to validate the code. Chapter 2 introduces a PML for the BGK model along the lines of [11]. Chapters 3 and 4 represent the heart of this work. In Chapter 3 we establish appropriate stability conditions for the BGK model with the PML. This is done via the symbol of the differential system and continued fraction expansions of the characteristic polynomial. In Chapter 4 we present a machinery based on the ANOVA expansion to systematically explore the parameter space. Finally, some relevant articles and side aspects, not discussed in the main text, are left to the appendices.

1

The BGK model

1.1 Arising from physics

The Bhatnagar-Gross-Krook model (hereafter BGK model) is an approximation to the Boltzmann equation defined as:

$$\frac{\partial f}{\partial t} + \boldsymbol{\zeta} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f = -\frac{1}{\gamma} \left(f - f_{\mathrm{B}}(\rho, \boldsymbol{u}) \right), \tag{1.1}$$

where $f \equiv f(t, \boldsymbol{\zeta}, \boldsymbol{x})$ represents the particle distribution function, $\boldsymbol{\zeta} = [\zeta_1, \zeta_2, \zeta_3]$ is the microscopic velocity, and γ is a relaxation time. Moreover, f_{B} denotes the Maxwell-Boltzmann equilibrium distribution function:

$$f_{\mathrm{B}}(\rho, \boldsymbol{u}) = \frac{\rho}{(2\pi\mathbb{R}\mathbb{T})^{d/2}} \exp\left(-\frac{|\boldsymbol{\zeta} - \boldsymbol{u}|^2}{2\mathbb{R}\mathbb{T}} \right),$$

where ρ and $\boldsymbol{u}$ are the macroscopic density and velocity, $\mathbb{R}$ is the gas constant, $\mathbb{T}$ is the thermodynamic temperature and d is the number of space dimensions. The two terms on the left-hand side of (1.1) represent mixing and transport of the particles, respectively, while the right-hand side takes into account the collisions between the particles.

The relationships between $f(t, \boldsymbol{\zeta}, \boldsymbol{x})$ and the macroscopic quantities, i.e. density ρ, momentum $\rho\boldsymbol{u}$ and pressure tensor P_{ij}, are given as [15]

$$\rho = \int_{-\infty}^{+\infty} f \, \mathrm{d}\boldsymbol{\zeta}, \qquad \rho u_i = \int_{-\infty}^{+\infty} \zeta_i f \, \mathrm{d}\boldsymbol{\zeta}, \qquad P_{ij} = \int_{-\infty}^{+\infty} (\zeta_i - u_i)(\zeta_j - u_j) f \, \mathrm{d}\boldsymbol{\zeta}. \tag{1.2}$$

The stress tensor σ_{ij} is defined by

$$\sigma_{ij} = p\, I - P_{ij}, \tag{1.3}$$

where $p = \frac{1}{3} \operatorname{tr}\{P_{ij}\} = \mathbb{R}\mathbb{T}\rho$ is the scalar pressure.

1.2 The approximation of the BGK model

Directly solving the BGK approximation to the Boltzmann equation being too demanding, we approximate it with an expansion in Hermite polynomials[1].

[1] Hermite polynomials are a type of classical orthogonal polynomial sequence. They show up in many applications, ranging from finite element methods (as shape functions of beams) to quantum

We expand the particle distribution function in a basis $\xi_k(\boldsymbol{\zeta})$ made up of Hermite polynomials:

$$f(t,\boldsymbol{\zeta},\boldsymbol{x}) = \frac{\rho}{(2\pi\mathbb{R}\mathbb{T})^{d/2}} \exp\left(-\frac{\boldsymbol{\zeta}\cdot\boldsymbol{\zeta}}{2\mathbb{R}\mathbb{T}}\right) \sum_{k=0}^{\infty} a_k(\boldsymbol{x},t)\,\xi_k(\boldsymbol{\zeta}).$$

After some manipulations, we find the *approximate form of the BGK model*:

$$\frac{\partial \boldsymbol{a}}{\partial t} + A_1\frac{\partial \boldsymbol{a}}{\partial x_1} + A_2\frac{\partial \boldsymbol{a}}{\partial x_2} = S(\boldsymbol{a}), \tag{1.4}$$

where $\boldsymbol{a} = (a_0, a_1, a_2, a_3, a_4, a_5)^T$ is the vector collecting the expansion coefficients and

$$A_1 = \sqrt{\mathbb{R}\mathbb{T}}\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad A_2 = \sqrt{\mathbb{R}\mathbb{T}}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \sqrt{2} \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 0 & 0 \end{bmatrix}.$$

Here $S(\boldsymbol{a})$ is a nonlinear source vector:

$$S(\boldsymbol{a}) = -\frac{1}{\gamma}\left(0,0,0,a_3 - \frac{a_1a_2}{a_0}, a_4 - \frac{a_1^2}{\sqrt{2}a_0}, a_5 - \frac{a_2^2}{\sqrt{2}a_0}\right)^T.$$

Note that $S(\boldsymbol{a})$ collects nonlinear terms, but also some linear, low-order terms. This means that it can be split as:

$$S(\boldsymbol{a}) = S_{\mathrm{L}}(\boldsymbol{a}) + S_{\mathrm{NL}}(\boldsymbol{a})$$

where

$$S_{\mathrm{L}}(\boldsymbol{a}) = -\frac{1}{\gamma}\,(0,0,0,1,1,1)\,\boldsymbol{a},$$

$$S_{\mathrm{NL}}(\boldsymbol{a}) = -\frac{1}{\gamma}\left(0,0,0,-\frac{a_1a_2}{a_0}, -\frac{a_1^2}{\sqrt{2}a_0}, -\frac{a_2^2}{\sqrt{2}a_0}\right)^T,$$

so we can rewrite (1.4) highlighting the nonlinear terms

$$\frac{\partial \boldsymbol{a}}{\partial t} + A_1\frac{\partial \boldsymbol{a}}{\partial x_1} + A_2\frac{\partial \boldsymbol{a}}{\partial x_2} - S_{\mathrm{L}}(\boldsymbol{a}) = S_{\mathrm{NL}}(\boldsymbol{a}). \tag{1.5}$$

physics (as eigenstates of the quantum harmonic oscillator). They also appear in the Hermite functions, which form a complete orthogonal system for the Fourier transform.

1.3 Hyperbolicity

In this section we look at system (1.4) and investigate its hyperbolic properties following previous work [7, 18]. Let P be the differential operator of the system (1.4), defined as

$$P(\partial/\partial \boldsymbol{x}) := -\left[A_1 \frac{\partial}{\partial x_1} + A_2 \frac{\partial}{\partial x_2}\right]. \tag{1.6}$$

In the most general case, P is also a function of $\boldsymbol{x} \equiv (x_1, x_2)$ and t, but here, since we are dealing with a system of PDEs with constant coefficients, it is only a function of $\partial/\partial \boldsymbol{x} \equiv (\partial/\partial x_1, \partial/\partial x_2)$. We define the *principal part* P_1 of the differential system by replacing the partial derivatives $\partial/\partial \boldsymbol{x}$ in P with $\boldsymbol{n} \equiv (n_1, n_2) \in \mathbb{R}^2$:

$$P_1(\boldsymbol{n}) := -\sqrt{\mathbb{R}\mathbb{T}} \begin{bmatrix} 0 & n_1 & n_2 & 0 & 0 & 0 \\ n_1 & 0 & 0 & n_2 & \sqrt{2}\, n_1 & 0 \\ n_2 & 0 & 0 & n_1 & 0 & \sqrt{2}\, n_2 \\ 0 & n_2 & n_1 & 0 & 0 & 0 \\ 0 & \sqrt{2}\, n_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2}\, n_2 & 0 & 0 & 0 \end{bmatrix}. \tag{1.7}$$

Again, in the most general case, P_1 would also be a function of $\boldsymbol{x}$ and t. In the following we show some properties of system (1.4), starting from the following definition.

Definition 1.1. *The system* (1.4) *is called* hyperbolic *if the* $m \times m$ *matrix* $P_1(\boldsymbol{x}, t, \boldsymbol{n})$ *is diagonalizable for each* $\boldsymbol{x}, \boldsymbol{n} \in \mathbb{R}^d,\ t \geq 0$.

Here m denotes the number of conserved variables, which in the BGK model is equal to 6, while d is the number of space dimensions, which in our case is equal to 2. One could also say that *the system* (1.4) *is hyperbolic if all the* m *eigenvalues of* $P_1(\boldsymbol{x}, t, \boldsymbol{n})$ *are real.* In our case, the eigenvalues of $P_1(\boldsymbol{n})$ are

$$0, \quad 0, \quad -\sqrt{n_1^2 + n_2^2}, \quad \sqrt{n_1^2 + n_2^2}, \quad -\sqrt{3}\sqrt{n_1^2 + n_2^2}, \quad \sqrt{3}\sqrt{n_1^2 + n_2^2}, \tag{1.8}$$

and, since they are all real, then (1.4) is indeed hyperbolic.

Definition 1.2. *The system of PDEs* (1.4) *is* symmetric hyperbolic *if each* A_i *is a symmetric matrix for each* $\boldsymbol{x} \in \mathbb{R}^d,\ t \geq 0$. *Moreover,* (1.4) *is* strictly hyperbolic *if for each* $\boldsymbol{x}, \boldsymbol{n} \in \mathbb{R}^d,\ \boldsymbol{n} \neq 0$ *and each* $t \geq 0$ *the matrix* $P_1(\boldsymbol{x}, t, \boldsymbol{n})$ *has* m *distinct real eigenvalues.*

We note that in our case the two matrices A_1 and A_2, which contain constant coefficients, are both symmetric and hence it is clear that (1.4) is a *symmetric* hyperbolic system. However, (1.4) is not *strictly* hyperbolic, since zero is appearing twice as eigenvalue of $P(\boldsymbol{n})$. We refer the reader to Appendix A for a proof of the well-posedness of the BGK model.

The hyperbolicity condition is important because it is equivalent to requiring that there are m distinct plane wave solutions of (1.4) for each direction $\boldsymbol{n}$ [7].

In general, hyperbolic systems may have eigenvalues that are zero and pairs of eigenvalues that have the same magnitude, but opposite signs. If we have, say, p pairs of eigenvalues that differ in their sign only, this implies that at each boundary we can impose at most p boundary conditions. In our case the eigenvalues in (1.8) are telling us that there are six characteristics, among which two are positive, two are negative and two are zero. According to our previous observation, this means that at any boundary we do not have to impose more than two boundary conditions, but we may impose an additional two corresponding to the non-propagating modes.

1.4 Relationship with the Navier-Stokes equations

Using the relationships in (1.2) and (1.3) and the properties of Hermite polynomials, one can find the following connections between the expansion coefficients $\boldsymbol{a}$ and the macroscopic quantities:

$$\rho = \int_{-\infty}^{+\infty} f_{\mathrm{B}}\,\mathrm{d}\boldsymbol{\zeta} = a_0, \quad u = \int_{-\infty}^{+\infty} \zeta_1 f_{\mathrm{B}}\,\mathrm{d}\boldsymbol{\zeta} = \frac{a_1\sqrt{\mathbb{R}\mathbb{T}}}{a_0}, \quad v = \int_{-\infty}^{+\infty} \zeta_2 f_{\mathrm{B}}\,\mathrm{d}\boldsymbol{\zeta} = \frac{a_2\sqrt{\mathbb{R}\mathbb{T}}}{a_0},$$

$$\sigma_{11} = -\int_{-\infty}^{+\infty} (\zeta_1 - u)^2 f_{\mathrm{B}}\,\mathrm{d}\boldsymbol{\zeta} + \mathbb{R}\mathbb{T}\,\rho = -\mathbb{R}\mathbb{T}\left(\sqrt{2}a_4 - \frac{a_1^2}{a_0}\right),$$

$$\sigma_{22} = -\int_{-\infty}^{+\infty} (\zeta_2 - v)^2 f_{\mathrm{B}}\,\mathrm{d}\boldsymbol{\zeta} + \mathbb{R}\mathbb{T}\,\rho = -\mathbb{R}\mathbb{T}\left(\sqrt{2}a_5 - \frac{a_2^2}{a_0}\right),$$

$$\sigma_{12} = -\int_{-\infty}^{+\infty} (\zeta_1 - u)(\zeta_2 - v) f_{\mathrm{B}}\,\mathrm{d}\boldsymbol{\zeta} + \mathbb{R}\mathbb{T}\,\rho = -\mathbb{R}\mathbb{T}\left(a_3 - \frac{a_1 a_2}{a_0}\right),$$

and the reverse:

$$a_0 = \rho, \quad a_1 = \frac{u\rho}{\sqrt{\mathbb{R}\mathbb{T}}}, \quad a_2 = \frac{v\rho}{\sqrt{\mathbb{R}\mathbb{T}}}, \tag{1.9}$$

$$a_3 = \frac{uv\rho - \sigma_{12}}{\sqrt{\mathbb{R}\mathbb{T}}}, \quad a_4 = \frac{\sqrt{2}}{2}\frac{u^2\rho - \sigma_{11}}{\sqrt{\mathbb{R}\mathbb{T}}}, \quad a_5 = \frac{\sqrt{2}}{2}\frac{v^2\rho - \sigma_{22}}{\sqrt{\mathbb{R}\mathbb{T}}}.$$

It is possible to show that from (1.4) one can recover the Navier-Stokes equations, under the assumptions that the relaxation time and the Mach number go to zero, namely in the case of weakly compressible flows only [15]. Let us consider three time scales γ, Γ_0, Γ_1 with the relation $\gamma \ll \Gamma_0 \ll \Gamma_1$. Here γ is of the order of magnitude of the collision time, Γ_0 represents an intermediate time scale, small enough to allow to consider the macroscopic quantities constant in time, and Γ_1 is the macroscopic time scale on which variations in density and momentum appear. On the scale Γ_0 under the condition that γ is very small, the coefficients (a_0, a_1, a_2) can be considered constant in time. Moreover, one can get a relation between the stresses and the flow field via a kinematic viscosity $\nu = \mathbb{R}\mathbb{T}\,\gamma$ and the ideal gas law $p = \mathbb{R}\mathbb{T}\,\rho$. The coefficients (a_3, a_4, a_5) are related to the macroscopic variables as

$$a_3 = -\gamma\left(\frac{\partial \rho v}{\partial x_1} + \frac{\partial \rho u}{\partial x_2}\right) + \frac{uv\rho}{\mathbb{R}\mathbb{T}},$$

$$a_4 = -\gamma\sqrt{2}\,\frac{\partial \rho u}{\partial x_1} + \frac{u^2\rho}{\sqrt{2}\mathbb{R}\mathbb{T}}, \tag{1.10}$$

$$a_5 = -\gamma\sqrt{2}\,\frac{\partial \rho v}{\partial x_2} + \frac{v^2\rho}{\sqrt{2}\mathbb{R}\mathbb{T}},$$

Substituting (1.9) and (1.10) into the first three equations of the BGK model (1.4), one can find

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x_1} + \frac{\partial \rho v}{\partial x_2} = 0,$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2}{\partial x_1} + \frac{\partial \rho uv}{\partial x_2} = \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} - \frac{\partial p}{\partial x_1},$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho uv}{\partial x_1} + \frac{\partial \rho v^2}{\partial x_2} = \frac{\partial \sigma_{21}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} - \frac{\partial p}{\partial x_2},$$

with the stress tensor being

$$\sigma_{ij} = \mathbb{R}\mathbb{T}\,\gamma\left(\frac{\partial \rho u_i}{\partial x_j} + \frac{\partial \rho u_j}{\partial x_i}\right).$$

We recognize the above equations as the two-dimensional isentropic Navier-Stokes equations for a weakly compressible flow.

1.5 Implementation aspects

Hereafter we will describe some aspects of the implementation and testing of the BGK model. To check whether this model is actually capable of reproducing the behaviour of some simple flows, we consider the Couette-Poiseuille flow. This problem is nice because we have an analytic solution to it. In fact, it is an exact solution in closed-form to the Navier-Stokes equations.

1.5.1 Spatial discretization

We decided to adopt a fourth order finite difference method for the spatial discretization, so as to have a completely regular grid. A scheme making use of more general, unstructured grids will be a bit more cumbersome to handle with, for instance when we would like to compare two solutions, since they would be living on two different grids and we would have to introduce some sort of mapping to allow the comparison to be performed. Moreover, if we have an unstructured grid, also periodic boundary conditions are more complicated to handle. However, in this project our main goal is not the spatial discretization but to understand the PML method. Therefore we regard a finite difference scheme suitable for this purpose, at least at this early stage.

Given a univariate function $f(x)$, the *fourth order accurate finite difference estimate of the first derivative* of f at x_i is

$$f'(x_i) = \frac{f(x_{i-2}) - 8f(x_{i-1}) + 8f(x_{i+1}) - f(x_{i+2})}{12\Delta x} + \mathcal{O}(\Delta x^4).$$

To see how this applies to our case, let us consider for instance the first equation in the BGK model (1.5):

$$\frac{\partial a_0}{\partial t} = -\sqrt{\mathbb{RT}}\left(\frac{\partial a_1}{\partial x} + \frac{\partial a_2}{\partial y}\right),$$

which becomes, after spatial discretization:

$$\frac{\partial a_0}{\partial t} = -\sqrt{\mathbb{RT}}\left(\frac{a_1(x_{i-2}, y_j) - 8a_1(x_{i-1}, y_j) + 8a_1(x_{i+1}, y_j) - a_1(x_{i+2}, y_j)}{12\Delta x} + \frac{a_2(x_i, y_{j-2}) - 8a_2(x_i, y_{j-1}) + 8a_2(x_i, y_{j+1}) - a_2(x_i, y_{j+2})}{12\Delta y}\right).$$

Note that the $\boldsymbol{a}$ are functions of two spatial variables, so it should be evident that when we take the derivative of $\boldsymbol{a}$ with respect to x we fix our attention on the same y_j and, conversely, when we take the derivative of $\boldsymbol{a}$ with respect to y we fix our attention on the same x_i. Moreover, note that here we have not done the time discretization yet. This makes the last equation an instance of the so-called scheme in *semi-discrete form*.

Figure 1.1 shows the spatial discretization stencil of the fourth order finite difference scheme adopted in this work. The bold $\boldsymbol{a}$ denotes the variables that are evolved according to the BGK model (1.5). If we repeat this spatial discretization

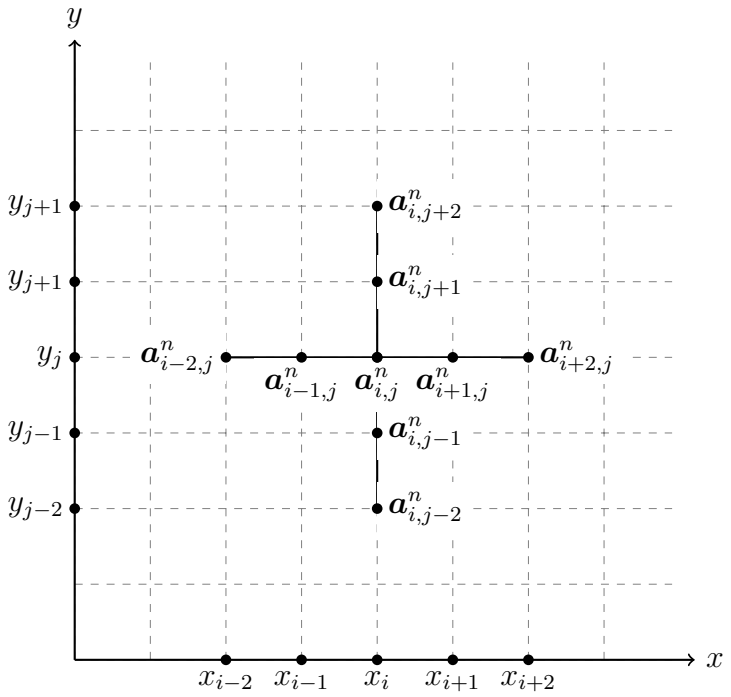


Figure 1.1: Stencil at time t^n.

for all the equations of the BGK model, then we get the semi-discrete form of all the model, and we can synthetically write

$$\frac{\partial \boldsymbol{a}}{\partial t} = \boldsymbol{f}\left(t,\ \boldsymbol{a}\right).$$

where $\boldsymbol{f}$ denotes the right-hand side of the BGK equations after spatial discretization.

1.5.2 Time discretization

For the time discretization, we employ a fourth order *Runge-Kutta method*:

$$\begin{aligned}
\boldsymbol{\kappa}_1 &= \boldsymbol{f}\left(t^n,\ \boldsymbol{a}^n\right), \\
\boldsymbol{\kappa}_2 &= \boldsymbol{f}\left(t^n + \frac{\Delta t}{2},\ \boldsymbol{a}^n + \frac{\Delta t}{2}\boldsymbol{\kappa}_1\right), \\
\boldsymbol{\kappa}_3 &= \boldsymbol{f}\left(t^n + \frac{\Delta t}{2},\ \boldsymbol{a}^n + \frac{\Delta t}{2}\boldsymbol{\kappa}_2\right), \\
\boldsymbol{\kappa}_4 &= \boldsymbol{f}\left(t^n + \Delta t,\ \boldsymbol{a}^n + \Delta t\boldsymbol{\kappa}_3\right), \\
\boldsymbol{a}^{n+1} &= \boldsymbol{a}^n + \frac{\Delta t}{6}\left(\boldsymbol{\kappa}_1 + 2\boldsymbol{\kappa}_2 + 2\boldsymbol{\kappa}_3 + \boldsymbol{\kappa}_4\right),
\end{aligned}$$

where $\boldsymbol{f}$ denotes the right-hand side of the BGK equations after spatial discretization, Δt is the time-step and $\boldsymbol{a}^n$ are the BGK variables computed at time t^n. The Runge-Kutta method is a multi-stage method in which each intermediate stage is in some sense equivalent to a forward Euler method [14].

1.5.3 The CFL condition

In general, for a d-dimensional space, the stability condition for a hyperbolic system is expressed as [23]

$$\Delta t \leq C \min\left(\frac{\Delta x_i}{\sqrt{d}|v|}\right),$$

where C is a constant that depends on the method, $i = 1, 2, \ldots, d$ and $|v| = (\sum_{i=1}^{d} v_i^2)^{1/2}$. In our case, the stable time-step becomes:

$$\Delta t \leq \frac{\Delta x}{\sqrt{3}\sqrt{2}\sqrt{2\mathbb{RT}}} = \frac{\Delta x}{2\sqrt{3\mathbb{RT}}},$$

where $\sqrt{3}$ is the largest eigenvalue, $\sqrt{2}$ is the square root of the number of space dimensions and $\sqrt{2\mathbb{RT}}$ is the largest entry of A_1 and A_2.

The stability condition in two space dimensions can be viewed as an extension of the well known result in 1D: the numerical domain of dependence of a time-dependent PDE has to contain the physical domain of dependence [21]. Courant, Friedrichs and Lewy wrote a fundamental paper in 1928 that was the first paper on the stability and convergence of finite difference methods for PDEs. Figure 1.2 provides a sketch of this concept for a problem in two space dimensions.

1.6 Imposing boundary conditions

In hyperbolic problems, the treatment of boundary conditions is closely related to the *theory of characteristics*. The characteristics are lines along which the information

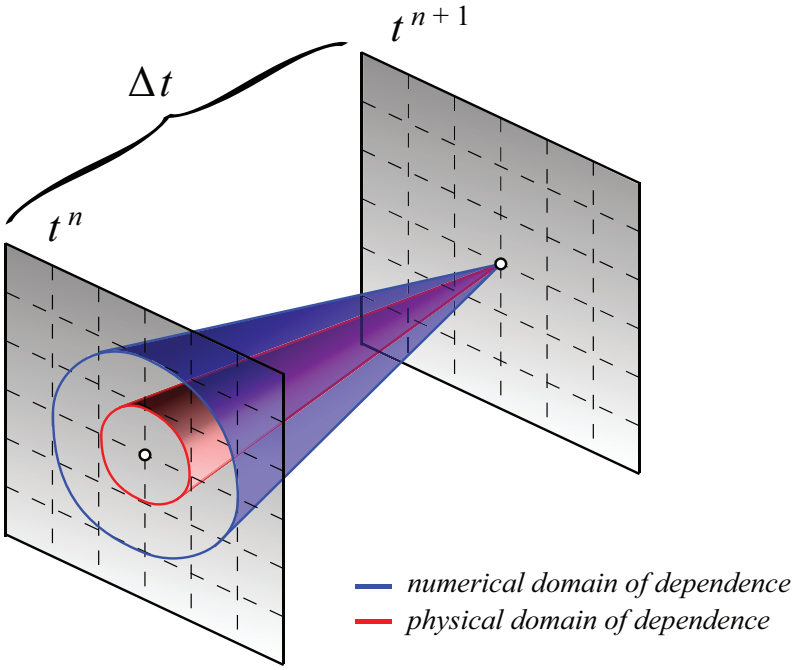


Figure 1.2: Graphical representation of the CFL condition in two space dimensions.

coming from the boundaries of the domain or from the initial condition travels. We point out that boundary conditions must be imposed on those boundaries from where the wave originates, but not on boundaries towards which the wave is propagating. This means that one has to have at least a minimal knowledge about the physics of the problem.

In one space dimension, the original system of hyperbolic equations can be recast in terms of the characteristic variables. In 1D this is relatively straightforward to achieve because a wave can only propagate along two directions: the left or the right. So if the wave moves to the right, we must specify a boundary condition at the left-end of the domain, whereas if the wave moves to the left, we must specify a boundary condition at the right-end of the domain.

When we go to more than one space dimension, things can get quickly out of hand, because a wave now has infinitely-many directions of propagation. Rigorously, one should impose the boundary conditions on the characteristic variables. However, when dealing with systems that depend on more than one spatial variable, as in our case (1.5), we have to resort to symmetrization techniques. Nonetheless the reconstruction on the characteristic variables is beyond the scope of this project, so we give just a short presentation of symmetrization techniques in Appendix A.

1.6.1 The mirroring technique

We detail the technique used in our code to enforce the boundary conditions. We resort to a technique that we may call the mirroring principle, which makes use of ghost points. Without loss of generality, we illustrate this for a 1D setting.

To explain how Dirichlet boundary conditions are imposed, let us consider Figure 1.3, which sketches a solution profile close to a left boundary of a 1D domain.

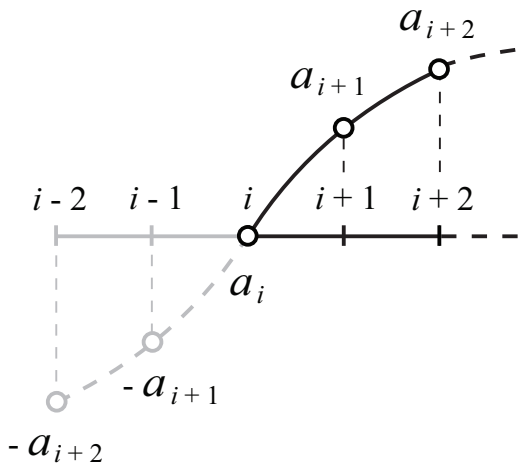


Figure 1.3: Mirroring technique for Dirichlet boundary conditions.

We look at the point i, located on the left boundary of the domain. Since, according to our fourth order finite difference scheme, we need four neighbouring points to compute the derivative at a point, we also look at the two inner grid-points $i+1, i+2$ closest to i, and create two ghost points at the left of i by using the same grid-size as in the inner grid. At these two ghost points $i-1$ and $i-2$ we assign the values of $-a_{i+1}$ and $-a_{i+2}$, respectively. One can view this as a central symmetry of the two points $(i+1, a_{i+1})$ and $(i+2, a_{i+2})$ with respect to the boundary point i. We do this to force the solution to pass through the boundary point i, thus satisfying the Dirichlet boundary condition. This technique, that we have illustrated in the case of homogeneous boundary conditions, extends also to the non-homogeneous case.

Now to see how this mirroring principle can be applied to the Neumann boundary conditions let us consider Figure 1.4.

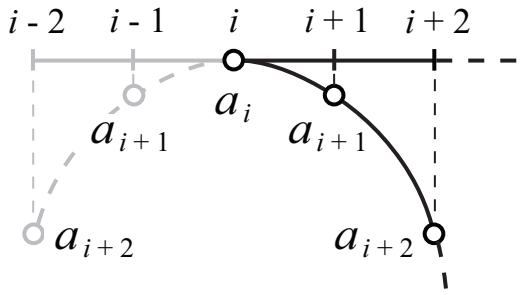


Figure 1.4: Mirroring technique for Neumann boundary conditions.

This time at the two ghost points $i-1$ and $i-2$ we assign the values of a_{i+1} and a_{i+2}, respectively. One can view this as an axial symmetry of the two points $(i+1, a_{i+1})$ and $(i+2, a_{i+2})$ with respect to a vertical axis passing through the boundary point i. This strategy forces the derivative of the solution at the boundary point i to be zero, thus satisfying a homogeneous Neumann boundary condition.

1.7 Flowchart of the BGK code

The flowcharts in this report are not intended to be exhaustive, since all the documentation is available within the code. At any rate, we hope that they can improve the understanding of the algorithm.

Figure 1.5 shows a flowchart of the code for the BGK model. The function `StartUpBGK` defines the computational domain, computes a stable time-step according to the CFL condition and sets the initial conditions for the BGK variables. Inside the loop the integration of the 2D BGK equations is performed until `FinalTime` is reached. The function `BGK_RHS2D` imposes the boundary conditions and evaluates the right-hand side of the BGK equations using the 4th order accurate centered finite difference scheme. The function `PlotBGK2D` is optional and allows us to visualize the time evolution of either the BGK variables or the physical variables.

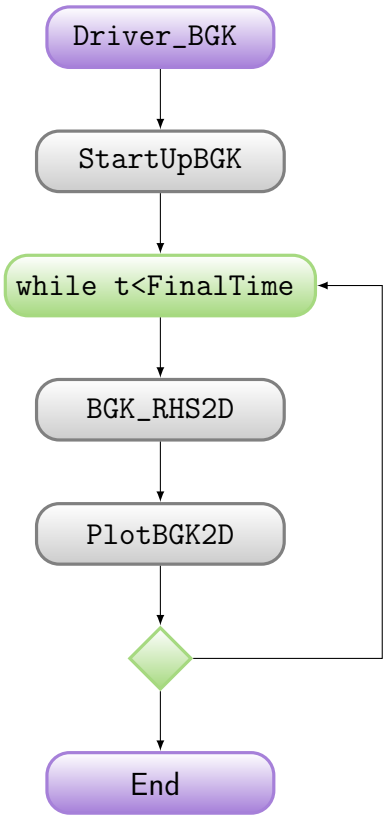


Figure 1.5: Flowchart of the code for the BGK model.

1.8 Accuracy test

To check that the numerical scheme is actually fourth order accurate we are going to test it for a simple Couette-Poiseuille flow. We exploit the fact that for the Couette-Poiseuille flow we have an exact solution to the Navier-Stokes equations.

Of course here we are not solving the Navier-Stokes equations, but this remains a good test for the code because, as we have seen above, there is a connection between the Navier-Stokes equations and the BGK model. Hence we use the exact solution for the Couette-Poiseuille flow into the BGK and we expect that the code should be able to maintain that solution. This can also be verified qualitatively.

We consider a square domain $L_x = L_y = 1.00$ with 20 grid-points in each direction. The values of the BGK parameters are

$$\mathbb{RT} = 1, \quad \nu = 0.01, \quad \gamma = \nu/\mathbb{RT}.$$

The initial conditions are set to

$$a_0 = \rho = \frac{p_{\text{in}} - x\,(p_{\text{in}} - p_{\text{out}})}{\mathbb{RT}}, \qquad u = 4u_{\max}\, y\,(1-y)\,, \qquad a_1 = \frac{u\rho}{\sqrt{\mathbb{RT}}},$$

$$a_2 = 0, \qquad a_3 = \gamma\, 4u_{\max}\, 2y\rho, \qquad a_4 = \frac{u^2\rho}{\sqrt{2}\mathbb{RT}} + \gamma u\sqrt{2}\,\frac{p_{\text{out}} - p_{\text{in}}}{\mathbb{RT}}, \qquad a_5 = 0.$$

Figure 1.6 shows the initial conditions for the BGK variables $\boldsymbol{a}$. Again we note that the initial conditions should be maintained throughout the simulation.

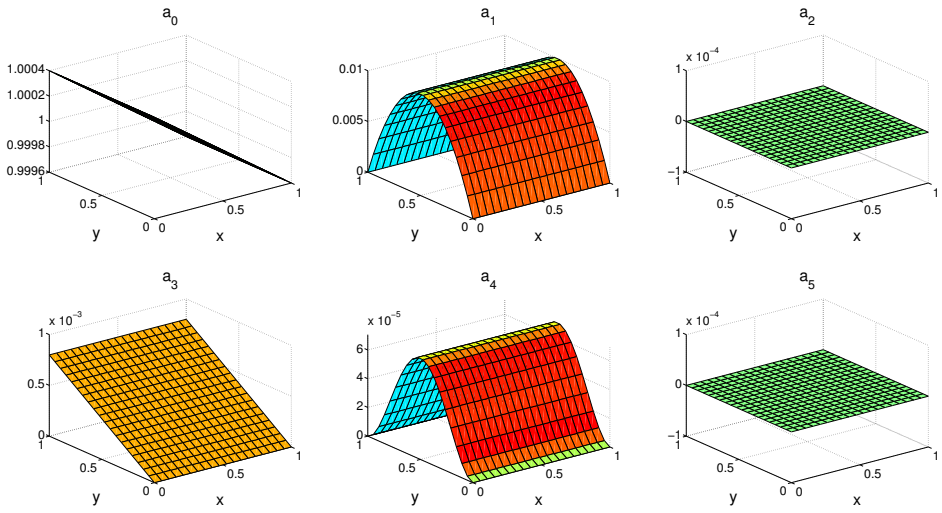


Figure 1.6: Initial conditions for the Couette-Poiseuille flow of the test.

Figure 1.7 shows the BGK variables $\boldsymbol{a}$ at the end of the simulation. We can observe that some minor oscillations arise, in particular for the coefficient a_2, but their magnitude is very small, of the order of 10^{-5}. On the one hand, from a qualitative point of view, we can conclude that the code is working properly for this benchmark case.

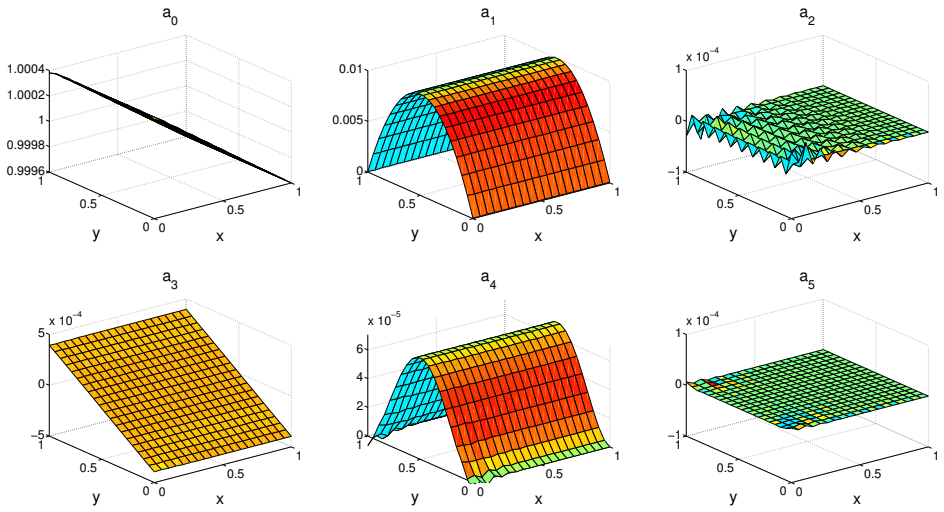


Figure 1.7: Solution of the BGK model at the end of the simulation, for the Couette-Poiseuille flow of the test.

On the other hand, in a more quantitative perspective, in order to know the actual order of convergence of the numerical scheme so implemented we can calculate the error between the exact solution and the numerical solution, at the final time `FinalTime`, according to different mesh-sizes $\Delta x = \Delta y$. We measure the error in the discrete L^2-norm since we are dealing with a smooth problem. The continuous L^2-norm of the error is

$$\|\varepsilon\|_{L^2} = \left(\iint_{\Omega} \left(u_{(x,y)} - u^{\text{ex}}_{(x,y)} \right)^2 \, \mathrm{d}x \, \mathrm{d}y \right)^{1/2},$$

which in discrete form turns into

$$\|\varepsilon\|_{L^2} = \left(\Delta x \Delta y \sum_i \sum_j \left(u_{ij} - u^{\text{ex}}_{ij} \right)^2 \right)^{1/2}.$$

with u_{ij} and u^{ex}_{ij} being the numerical solution and the exact solution, respectively, computed at the discrete grid-points.

Figure 1.8 shows the result of the accuracy test. It can be observed that the actual order of accuracy of the method is not exactly 4, but nearly 3. This is probably due to the fact that we are not imposing the boundary conditions on the characteristic variables. If we wanted to do this really accurately, we should resort to the techniques discussed in Appendix A.

It may also be possible that the order of accuracy of the method is not exactly 4 because the Couette-Poiseuille flow is an exact solution to the Navier-Stokes equations, but not to the BGK model. Thus the error in the convergence behaviour may not be due to the numerical scheme, but to the modelling. Further investigations are left for future work.

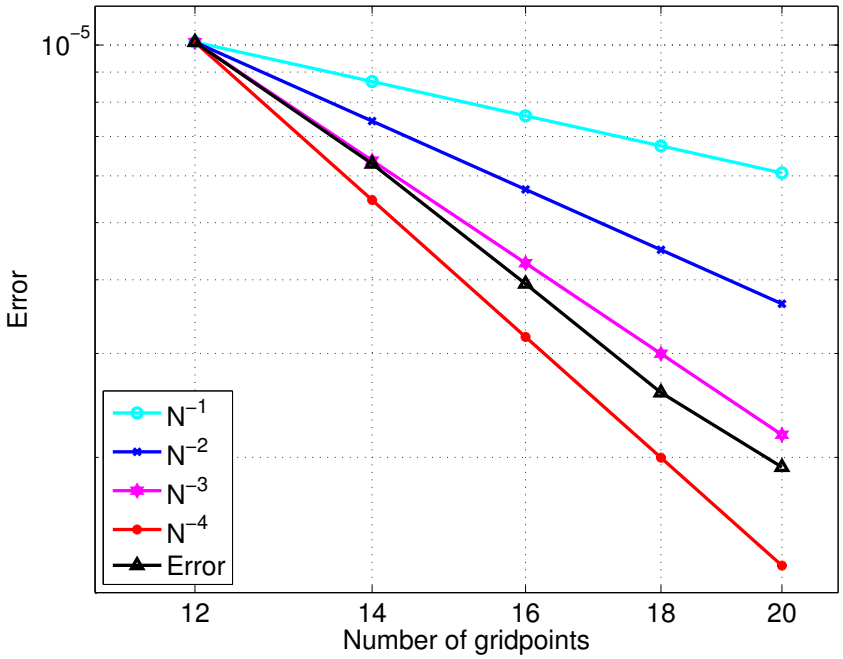


Figure 1.8: Accuracy test.

2

A PML for the BGK model

There is no branch of mathematics, however abstract, which may not some day be applied to phenomena of the real world.

– Lobachevsky

The PML technique was originally introduced in 1994 by Bérenger [3], who developed it by starting from physical considerations on electromagnetic waves. Bérenger modified Maxwell's equations so that waves getting into the absorbing layer decay without reflections at the interface. In his original formulation Bérenger adopted a splitting technique of the Maxwell's equations. Later it was shown that such splitting technique breaks the hyperbolicity of the system, leading to numerical instabilities in long time simulations.

In 2003, Hagstrom [17] proposed a new technique for developing PMLs for hyperbolic systems. This approach is based on the modal analysis in Laplace-Fourier space and it makes the solutions inside the PML decay as they propagate. Hagstrom's paper, along with the theory for developing PMLs for linear hyperbolic systems, is discussed in Appendix B.

By following in the steps of Hagstrom, Appelö et al. [2] developed a thorough mathematical analysis of PMLs for linear hyperbolic systems by providing general tools to establish stability and well-posedness.

2.1 The BGK+PML model

We emphasize that the approach of [17] and [2] is applicable only to *linear* hyperbolic systems and that our BGK model (1.5) contains a nonlinear term. Nonetheless, if the nonlinear term $S_{\mathrm{NL}}(\boldsymbol{a})$ appearing in (1.5) is neglected, then the BGK model is indeed a linear hyperbolic system and hence it is possible to apply the technique proposed by Hagstrom to construct a PML for this problem.

This path has actually been pursued by Gao et al. [11], who neglected the nonlinear term in (1.5), followed the approach proposed by Hagstrom to construct a PML for the BGK model[1], and finally appended the nonlinear term at the equations. Here we just give the final result they obtained.

[1] In the following, we will simply refer to the BGK model coupled with a PML as the *BGK+PML model*.

The full absorbing layer formulation for (1.4) is

$$\begin{cases} \dfrac{\partial \boldsymbol{a}}{\partial t} + A_1 \left(\dfrac{\partial \boldsymbol{a}}{\partial x_1} + \sigma_1 \left(\lambda_0 \boldsymbol{a} + \boldsymbol{\omega} \right) \right) + A_2 \left(\dfrac{\partial \boldsymbol{a}}{\partial x_2} + \sigma_2 \left(\tilde{\lambda}_0 \boldsymbol{a} + \boldsymbol{\theta} \right) \right) = S(\boldsymbol{a}), \\ \dfrac{\partial \boldsymbol{\omega}}{\partial t} + \alpha_1 \dfrac{\partial \boldsymbol{\omega}}{\partial x_2} + (\alpha_0 + \sigma_1)\boldsymbol{\omega} + \dfrac{\partial \boldsymbol{a}}{\partial x_1} + \lambda_0(\alpha_0 + \sigma_1)\boldsymbol{a} - \lambda_1 \dfrac{\partial \boldsymbol{a}}{\partial x_2} = \mathbf{0}, \\ \dfrac{\partial \boldsymbol{\theta}}{\partial t} + \tilde{\alpha}_1 \dfrac{\partial \boldsymbol{\theta}}{\partial x_1} + (\tilde{\alpha}_0 + \sigma_2)\boldsymbol{\theta} + \dfrac{\partial \boldsymbol{a}}{\partial x_2} + \tilde{\lambda}_0(\tilde{\alpha}_0 + \sigma_2)\boldsymbol{a} - \tilde{\lambda}_1 \dfrac{\partial \boldsymbol{a}}{\partial x_1} = \mathbf{0}, \end{cases} \tag{2.1}$$

where $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ are auxiliary variables and $\alpha_0, \lambda_0, \alpha_1, \lambda_1, \tilde{\alpha}_0, \tilde{\lambda}_0, \tilde{\alpha}_1, \tilde{\lambda}_1$ are some parameters. The damping functions in the x- and y-directions are σ_1 and σ_2, respectively.

2.2 Perfect matching

The key idea behind the PML is that the eigenfunctions for the eigenvalue problem *inside* the layer have to be the same as *outside* the layer. This is the most straightforward way to design a PML so that reflections at the PML interface are prevented [17]. In the following, we are going to verify that this is actually the case for the BGK+PML model (2.1).

We consider the homogeneous case of the first equation in (2.1) for a PML developing in the x_1-direction only, with σ_1 being a constant for the sake of simplicity. Hence the governing equation *outside the layer* is

$$\frac{\partial \boldsymbol{a}}{\partial t} + A_1 \frac{\partial \boldsymbol{a}}{\partial x_1} + A_2 \frac{\partial \boldsymbol{a}}{\partial x_2} = \mathbf{0},$$

with a Laplace-Fourier transform as

$$\left(sI + \lambda A_1 + \mathrm{i}k_2 A_2 \right) \boldsymbol{\phi}(x_1, \mathrm{i}k_2, s) = \mathbf{0}, \tag{2.2}$$

and the modal solution outside the layer being

$$\hat{\boldsymbol{a}} = e^{\lambda x_1} \boldsymbol{\phi}(x_1, \mathrm{i}k_2, s).$$

Inside the layer, the governing equation is

$$\frac{\partial \boldsymbol{a}}{\partial t} + A_1 \left(\frac{\partial \boldsymbol{a}}{\partial x_1} + \sigma_1 \left(\lambda_0 \boldsymbol{a} + \boldsymbol{\omega} \right) \right) + A_2 \frac{\partial \boldsymbol{a}}{\partial x_2} = \mathbf{0}. \tag{2.3}$$

This equation has been constructed on the basis of the following Ansatz for the modal solution inside the layer:

$$\hat{\boldsymbol{a}}_{\mathrm{PML}} = e^{\lambda x_1 + \left[\frac{\lambda - \lambda_1 \mathrm{i}k_2 + \lambda_0 \alpha_0}{s + \alpha_1 \mathrm{i}k_2 + \alpha_0} - \lambda_0 \right] \sigma_1 x_1} \boldsymbol{\phi}(x_1, \mathrm{i}k_2, s). \tag{2.4}$$

It can be shown (see Appendix C) that the Laplace-Fourier transform of (2.3) yields

$$\left(sI + A_1 \left(\left(I - \frac{\sigma_1}{\hat{r} + \sigma_1} \right) \left(\frac{\partial}{\partial x_1} + \sigma_1 \lambda_0 \right) + \frac{\sigma_1}{\hat{r} + \sigma_1} \left(\lambda_1 \mathrm{i}k_2 - \lambda_0 \alpha_0 \right) \right) + \mathrm{i}k_2 A_2 \right) \hat{\boldsymbol{a}}_{\mathrm{PML}} = \mathbf{0}.$$

By inserting the Ansatz for the modal solution inside the layer (2.4) into the last equation, and approaching the PML interface (i.e. letting $\sigma_1 \to 0$), one can see that equation (2.2) is recovered. This is exactly what we wanted: the eigenfunctions for the governing equations recast in the Laplace-Fourier space remain the same across the PML interface.

In all their simulations, Gao et al. [11] assumed the parameters as

$$\lambda_1 = 0, \quad \lambda_0 = 0, \quad \alpha_1 = 0, \quad \alpha_0 \neq 0,$$

$$\tilde{\lambda}_1 = 0, \quad \tilde{\lambda}_0 = 0, \quad \tilde{\alpha}_1 = 0, \quad \tilde{\alpha}_0 \neq 0.$$

The choice of these parameters, together with the fact that the precise role of each parameter is not well understood yet, leaves room for further study that we are going to pursue in this work.

2.3 Our study case

Hereinafter we will always consider a PML along the x_1-direction only, as in the example of Figure 1. This is equivalent to say that we set $\sigma_2 = 0$, so that the system (2.1) turns into

$$\begin{cases} \dfrac{\partial \boldsymbol{a}}{\partial t} + A_1 \left(\dfrac{\partial \boldsymbol{a}}{\partial x_1} + \sigma_1 \left(\lambda_0 \boldsymbol{a} + \boldsymbol{\omega} \right) \right) + A_2 \dfrac{\partial \boldsymbol{a}}{\partial x_2} = S(\boldsymbol{a}), \\ \dfrac{\partial \boldsymbol{\omega}}{\partial t} + \alpha_1 \dfrac{\partial \boldsymbol{\omega}}{\partial x_2} + (\alpha_0 + \sigma_1)\boldsymbol{\omega} + \dfrac{\partial \boldsymbol{a}}{\partial x_1} + \lambda_0(\alpha_0 + \sigma_1)\boldsymbol{a} - \lambda_1 \dfrac{\partial \boldsymbol{a}}{\partial x_2} = \boldsymbol{0}. \end{cases} \tag{2.5}$$

For the system (2.5), we report in Table 2.1 a summary of the parameters and their occurrence in the equations, which will turn out to be useful later.

Parameter	Occurrence
λ_0	Once in the $\boldsymbol{a}$ equation and once in the $\boldsymbol{\omega}$ equation, in both cases controlling the behaviour of the linear term in $\boldsymbol{a}$. We note that, in the $\boldsymbol{\omega}$ equation, λ_0 appears as a coefficient of $\boldsymbol{a}$ only if $\alpha_0 \neq 0$.
λ_1	Once in the $\boldsymbol{\omega}$ equation, as a multiplying coefficient of the derivative of $\boldsymbol{a}$ with respect to x_2.
α_0	Twice in the $\boldsymbol{\omega}$ equation, the first time as a coefficient of the linear term in $\boldsymbol{\omega}$ and the second time as a coefficient of the linear term in $\boldsymbol{a}$. We note that, in the $\boldsymbol{\omega}$ equation, α_0 appears as a coefficient of $\boldsymbol{a}$ only if $\lambda_0 \neq 0$.
α_1	Once in the $\boldsymbol{\omega}$ equation, as a multiplying coefficient of the derivative of $\boldsymbol{\omega}$ with respect to x_2.

Table 2.1: Occurrence of the parameters of the BGK+PML model (2.5).

2.4 Damping functions

The $\sigma_1 \geq 0$ and $\sigma_2 \geq 0$ appearing in (2.1) are the damping functions, which are assumed to be smooth and equal to zero at the PML interface. In general we assume a damping function σ of the form

$$\sigma(x) = C \left(\frac{x - x_0}{L} \right)^{\beta},$$

where x_0 represents the abscissa at which the PML begins, L is the thickness of the layer, and the exponent β is used to control the smoothness of the absorption profile. The constant C represents the overall strength of the absorption and it is usually chosen as the inverse of the time-step $C \simeq (\Delta t)^{-1}$ to avoid restriction on the time-step caused by the PML. Figure 2.1 illustrates the shape of the damping function σ_1 on the computational domain of the BGK+PML problem.

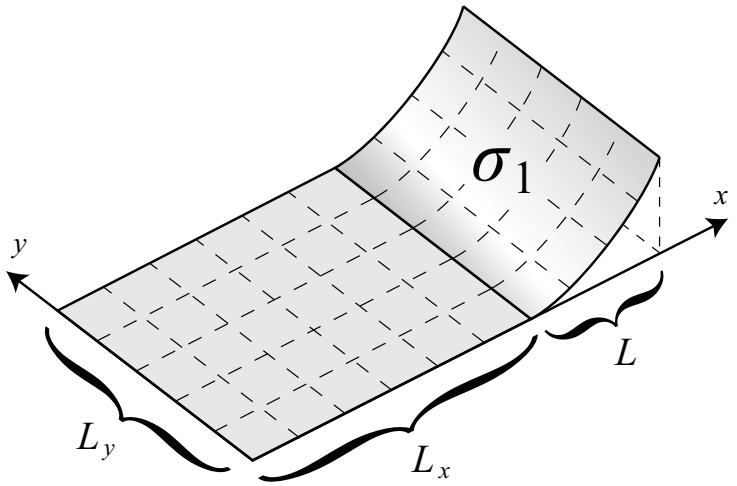


Figure 2.1: Illustration of the damping function σ_1.

2.5 The role of $S(\boldsymbol{a})$

In constructing equations (2.1), Gao et al. [11] neglected the whole of $S(\boldsymbol{a})$, but it turns out that even when taking into account the linear part and following the same steps, one recovers the same equations. In other words, if instead of looking at the homogeneous system we take the inhomogeneous system with just the linear terms, then in fact the theory in [2] can already treat this and we get the same equations.

The next step for an improvement would be to linearize the term $S_{\mathrm{NL}}(\boldsymbol{a})$. In fact it turns out that, if the time-step Δt is small enough, then one can assume that the variables $\boldsymbol{a}$ of the BGK model are constant over Δt. In other words, when we go from t^n to t^{n+1}, we keep the values of $\boldsymbol{a}$ locally constant in time, and we just use $\boldsymbol{a}^n$. This is what we have done in our code: when, at time t^{n+1}, we have to compute the stresses $\sigma_{11}, \sigma_{22}, \sigma_{12}$ that appear in $S(\boldsymbol{a})$, we calculate them by using the numerical solution $\boldsymbol{a}^n$ that we have already found at time t^n.

2.6 Flowchart of the BGK+PML code

The code for the BGK+PML model is similar to the one for the plain BGK, but here we also need to set up the PML parameters, the damping function σ_1 and to solve the additional equations for the auxiliary variables.

Figure 2.2 shows a flowchart for the BGK+PML code[2]. The function `StartUpBGKPML` defines the computational domain, computes a stable time-step, sets the initial conditions for the BGK coefficients, with a peak in the initial density distribution, and initializes the auxiliary variables and the damping functions of the PML. Inside the loop the integration of the 2D BGK+PML equations (2.5) is performed until `FinalTime` is reached. The function `BGKPML_RHS2D` imposes the boundary conditions and evaluates the right-hand of the BGK+PML equations using the 4th order accurate centered finite difference scheme. The function `PlotBGKPML2D` is optional and allows us to visualize the time evolution of either the BGK variables or the physical variables.

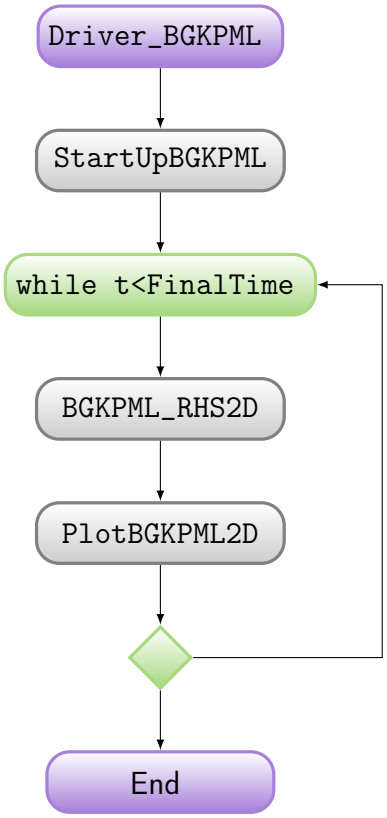


Figure 2.2: Flowchart of the BGK+PML code.

[2]There are some other subroutines that, for the sake of readability, are not reported here. At any rate, all the documentation can be found in the code.

2.7 Simulations

In this section we discuss and report the results of some simulations that have been carried out to assess to which extent the BGK+PML model (2.5) is capable of accurately reproducing the results of the plain BGK model. We first detail the parameters used in the simulations with the BGK+PML model, and point out the differences with respect to the simulations with the plain BGK model.

We consider a square domain $L_x = L_y = 1.00$, with an absorbing PML at the right-hand boundary and with wall boundary conditions on all the others. We have 20 grid-points in each direction, so that the mesh-size is equal to $\Delta x = \Delta y = 0.0526$.

We also need to decide the parameters of the damping function σ_1. In our implementation, the thickness of the PML is to be specified as a percentage of the width of the domain L_x. In this case $L_x = 1.00$ so the PML thickness factor coincides with the PML thickness, but in general this is not needed. For our simulations hereinafter we choose a PML thickness of $L = 0.40$. The exponent β appearing in the expression of σ_1 has been set to 4, while the overall absorption strength is calculated as $C = 1/\Delta t$. We note that σ_1 does not vary in time.

In order to generate a propagating wave in our domain we can perturb the initial density distribution. A reasonable choice is to assume an initial density distribution with a peak located at the centre of the domain:

$$a_0(x, y, t = 0) = 2\,(p_{\text{in}} - p_{\text{out}}) \exp\left[-\varepsilon\sqrt{(x - x_0)^2 + (y - y_0)^2}\right] + 1.00,$$

where

$$x_0 = L_x/2, \quad y_0 = L_y/2.$$

The factor ε at the exponent is set to $\varepsilon = 10$ to ensure a quick spatial decay of the peak, since we want all of its support to be outside the PML.

At the initial time, the velocity and the auxiliary variables $\boldsymbol{\omega}$ are set equal to zero everywhere in the domain.

For the BGK model without the PML, we choose to keep exactly the same data, with the only exception of L_x^{BGK} being approximately $2.5 \times L_x$ (the exact value depends on the mesh-size Δx). This is done because we have to guarantee that any wave propagating in the positive x-direction has sufficient space to propagate. At any rate, the additional computational cost if we want to simulate a decaying wave without the PML is clear.

In all of the following simulations, we keep the parameters of the PML equal to zero except for α_0 which is set to 1:

$$\lambda_1 = 0, \quad \lambda_0 = 0, \quad \alpha_1 = 0, \quad \alpha_0 = 1.$$

We choose the simulation time `FinalTime` to be 1.00, which is sufficient to allow the wave to enter into the PML, so that we can observe how the presence of the PML affects the simulation.

In the following pages we report the snapshots of the density distribution and the velocity field at different times. The coloured contours represent the density distribution, while the vectors depict the velocity field.

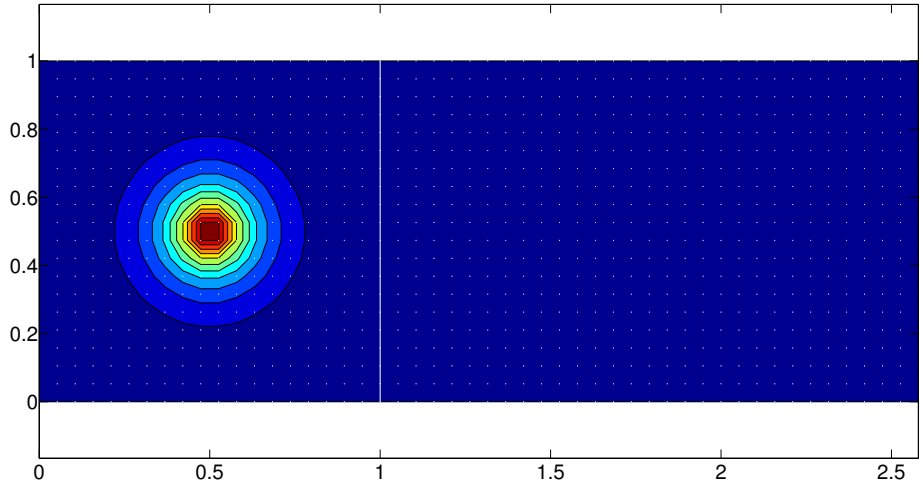
Figure 2.3 illustrates the initial conditions of the problem, for both the BGK and the BGK+PML models. We note that the peak in the initial density is entirely located in the original domain $L_x \times L_y$, outside the PML.

Figure 2.4 shows the visual outcome of the simulations at $t = 0.70$. At this time the wave has already entered into the PML, which in turns has already begun to damp out the wave. This can be readily observed by comparing Figures 2.4a and 2.4b. We note that the waves do not decay immediately as they enter into the PML, due to the shape of the damping function σ_1. In fact, the strength of the absorption becomes larger as the waves further penetrate into the PML. This is also highlighted by the velocity vectors shrinking down to points as one moves towards the end of the PML.

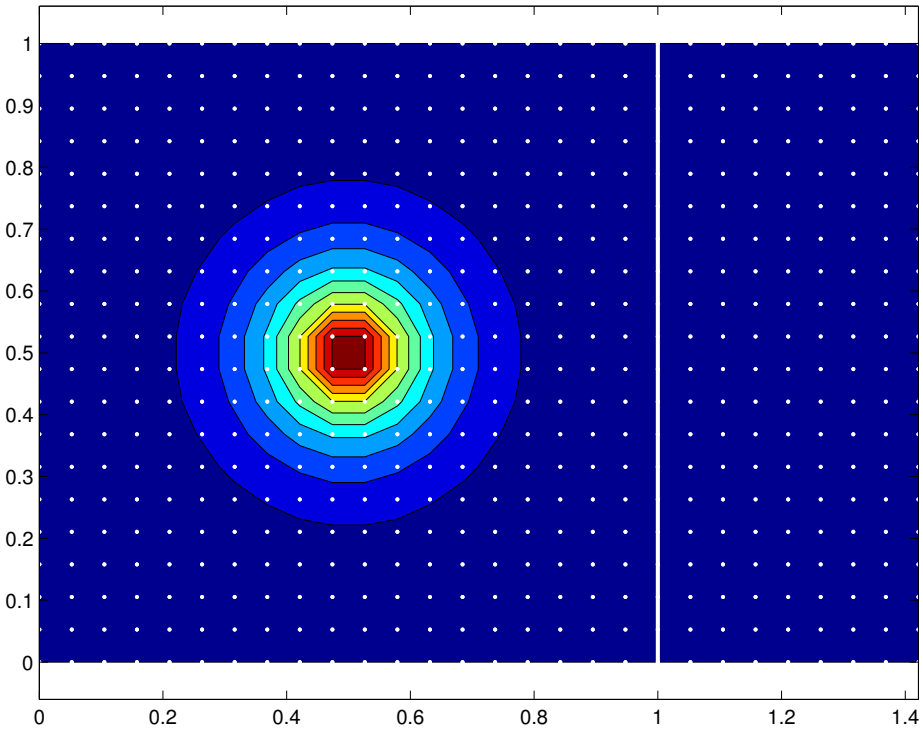
Figure 2.5 shows the final outcome of the simulations. Figure 2.5a emphasizes the presence of waves propagating to the right, while in Figure 2.5b those waves have been damped out thanks to the PML. If one compares the solution on the domain $L_x \times L_y$ in Figure 2.5a with the one in Figure 2.5b, one can conclude that from a qualitative point of view the BGK+PML model is behaving nicely. We can notice only minor differences and no significant signs of reflections. As expected, not only the waves entering in the PML have been absorbed, but most importantly they have not affected the solution on the original domain $L_x \times L_y$.

From Figures 2.3, 2.4 and 2.5 it appears that the BGK+PML model is qualitatively capable of reproducing the results obtained with the plain BGK. The more systematic and quantitative analysis are left for the remaining chapters.

As we have seen, the role and the importance of the parameters appearing in the BGK+PML model is not yet well understood. In what follows we will try to improve our understanding of the BGK+PML model (2.5), first by carrying out a stability analysis to establish reasonable bounds on the parameters, and then by performing a sensitivity analysis.

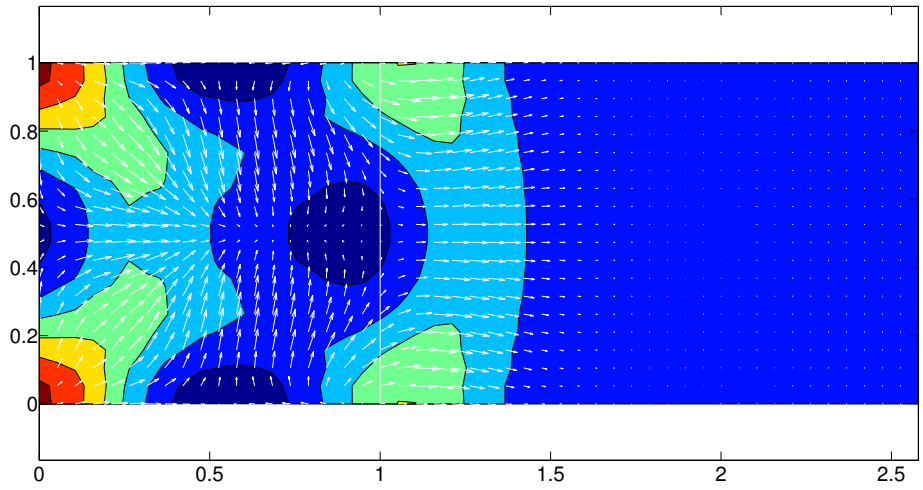


(a) BGK.

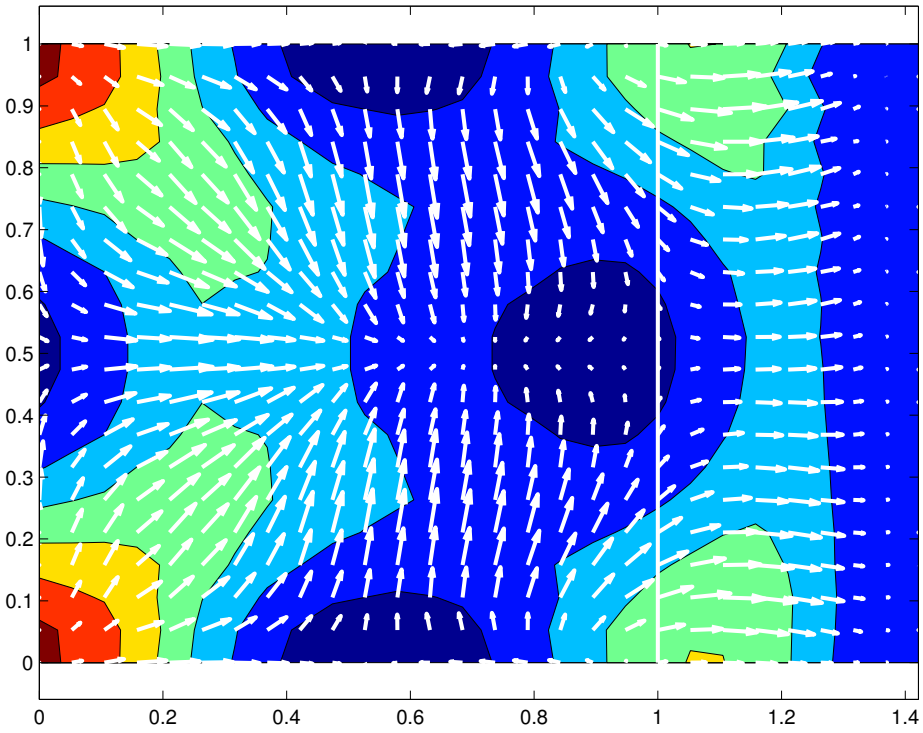


(b) BGK+PML.

Figure 2.3: Initial density distribution and velocity field.

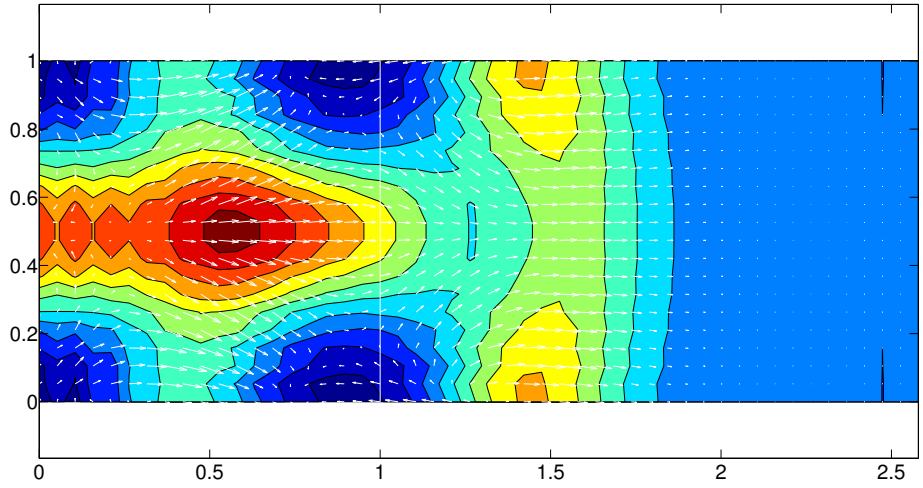


(a) BGK.

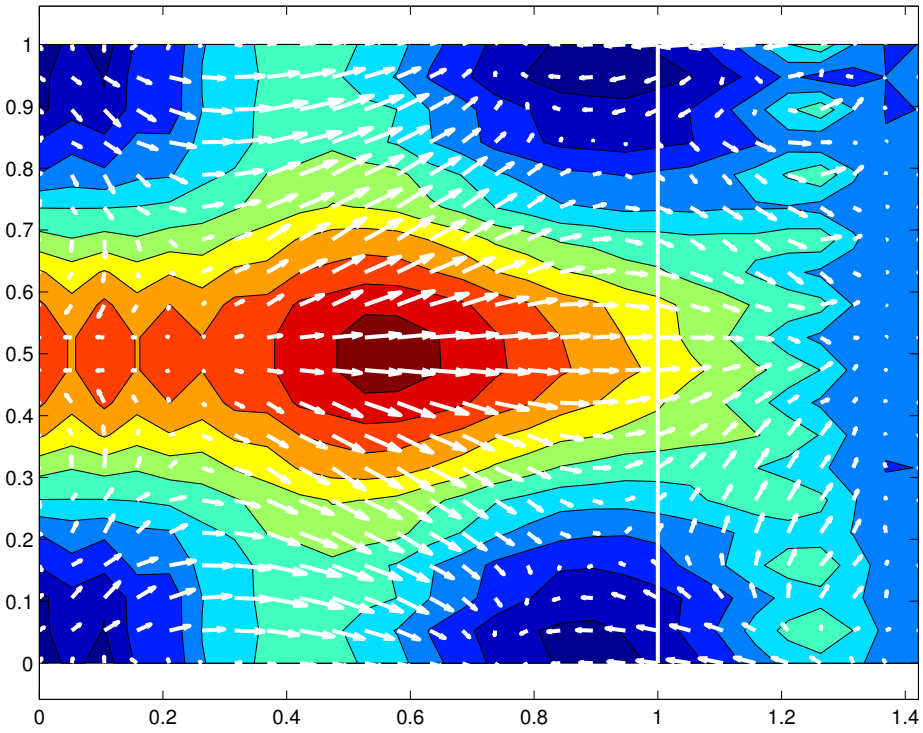


(b) BGK+PML.

Figure 2.4: Density distribution and velocity field at $t = 0.70$.



(a) BGK.



(b) BGK+PML.

Figure 2.5: Density distribution and velocity field at $t = 1.00$.

3

Stability analysis

> *Whenever possible, a problem should be analyzed and put into a proper form before it is run on a computer. An analysis is necessary to establish confidence in the alleged results. But analysis may also be valuable in that it can often establish early in the game ways of carrying out a computation which will save time. One good thought may be worth a hundred hours on the computer.*
>
> – Davis and Rabinowitz

In this chapter we are going to study the stability of the BGK+PML model through a couple of techniques to check stability of differential systems. In particular, we will analyse the problem by:

- enforcing energy decay;
- continued fraction expansion.

A key role in both of these methods is played by the *symbol* of the differential operator of the system. The analysis carried out in this chapter is going to provide us with some reasonable bounds on the parameters appearing in the BGK+PML model. These bounds will then be used in the following chapter to choose carefully the parameters for the simulations needed to perform a sensitivity analysis.

3.1 The symbol of the BGK+PML model

We recall that the governing equations of the BGK+PML model are:

$$\begin{cases} \dfrac{\partial \boldsymbol{a}}{\partial t} + A_1 \left(\dfrac{\partial \boldsymbol{a}}{\partial x_1} + \sigma_1 \left(\lambda_0 \boldsymbol{a} + \boldsymbol{\omega} \right) \right) + A_2 \left(\dfrac{\partial \boldsymbol{a}}{\partial x_2} + \sigma_2 \left(\tilde{\lambda}_0 \boldsymbol{a} + \boldsymbol{\theta} \right) \right) = S(\boldsymbol{a}), \\ \dfrac{\partial \boldsymbol{\omega}}{\partial t} + \alpha_1 \dfrac{\partial \boldsymbol{\omega}}{\partial x_2} + (\alpha_0 + \sigma_1)\boldsymbol{\omega} + \dfrac{\partial \boldsymbol{a}}{\partial x_1} + \lambda_0(\alpha_0 + \sigma_1)\boldsymbol{a} - \lambda_1 \dfrac{\partial \boldsymbol{a}}{\partial x_2} = \mathbf{0}, \\ \dfrac{\partial \boldsymbol{\theta}}{\partial t} + \tilde{\alpha}_1 \dfrac{\partial \boldsymbol{\theta}}{\partial x_1} + (\tilde{\alpha}_0 + \sigma_2)\boldsymbol{\theta} + \dfrac{\partial \boldsymbol{a}}{\partial x_2} + \tilde{\lambda}_0(\tilde{\alpha}_0 + \sigma_2)\boldsymbol{a} - \tilde{\lambda}_1 \dfrac{\partial \boldsymbol{a}}{\partial x_1} = \mathbf{0}. \end{cases} \tag{3.1}$$

This system can be rewritten in matrix form as

$$\frac{\partial}{\partial t}\begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{\omega}\\ \boldsymbol{\theta}\end{bmatrix} = -\begin{bmatrix} A_1\left(\frac{\partial}{\partial x_1}+\sigma_1\lambda_0\right)+A_2\left(\frac{\partial}{\partial x_2}+\sigma_2\tilde{\lambda}_0\right) & A_1\sigma_1 & A_2\sigma_2 \\ I\left(\frac{\partial}{\partial x_1}+\lambda_0(\alpha_0+\sigma_1)-\lambda_1\frac{\partial}{\partial x_2}\right) & I\left(\alpha_1\frac{\partial}{\partial x_2}+\alpha_0+\sigma_1\right) & O \\ I\left(\frac{\partial}{\partial x_2}+\tilde{\lambda}_0(\tilde{\alpha}_0+\sigma_2)-\tilde{\lambda}_1\frac{\partial}{\partial x_1}\right) & O & I\left(\tilde{\alpha}_1\frac{\partial}{\partial x_1}+\tilde{\alpha}_0+\sigma_2\right) \end{bmatrix}\begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{\omega}\\ \boldsymbol{\theta}\end{bmatrix}.$$

We refer to the matrix on the right-hand side as the *differential operator* of the system and we denote it by $P \equiv P(\partial/\partial x_1, \partial/\partial x_2)$. Moreover, we denote $[\boldsymbol{a},\ \boldsymbol{\omega},\ \boldsymbol{\theta}]^T$ by $\boldsymbol{u} \equiv \boldsymbol{u}(x_1, x_2, t)$. Hence we are dealing with a general system of the type:

$$\boldsymbol{u}_t = P\boldsymbol{u}, \tag{3.2}$$

whose initial condition can be expressed by:

$$\boldsymbol{u}(x_1, x_2, t=0) = \boldsymbol{f}(x_1, x_2).$$

The stability of such a problem can be studied by means of Fourier analysis (often called also Von Neumann analysis). In fact, if we perform a Fourier transform of (3.2) (or, equivalently, hypothesize a periodic solution, compute its Fourier series with complex coefficients, and insert it into (3.2)) we get:

$$\frac{\mathrm{d}\hat{\boldsymbol{u}}}{\mathrm{d}t} = \hat{P}\hat{\boldsymbol{u}}, \tag{3.3}$$

$$\hat{\boldsymbol{u}}(k_1, k_2, t=0) = \hat{\boldsymbol{f}}(k_1, k_2),$$

where $\hat{\boldsymbol{u}} \equiv \hat{\boldsymbol{u}}(k_1, k_2, t)$ are the *modes* and $\hat{P} \equiv \hat{P}(\mathrm{i}k_1, \mathrm{i}k_2)$ is called the *symbol* of the differential operator P. Equation (3.3) is a recasting of (3.2) in the frequency domain, with k_1, k_2 being the Fourier variables. Moreover, (3.3) is a system of ordinary differential equations with constant coefficients, having solution

$$\hat{\boldsymbol{u}} = e^{\hat{P}t}\hat{\boldsymbol{f}}(k_1, k_2). \tag{3.4}$$

This recasting in the frequency domain of our original problem (3.1) is useful because it turns a differential problem into an *algebraic* problem, and there are results based on the symbol $\hat{P}$ to establish well-posedness and stability (see, for instance, [16]).

Since we are dealing with a wave-dominated problem, it is reasonable to expect that the solution $\hat{\boldsymbol{u}}$ in (3.4) will be evolving in a decaying fashion with time. In the light of this, and because of the properties of the matrix exponential $e^{\hat{P}t}$, the following necessary condition for well-posedness should not be a surprise.

Theorem 3.1 (Petrovskii condition)**.** *A necessary condition for well-posedness of* (3.2) *is that, for all* k*, the eigenvalues* λ *of* $\hat{P}(\mathrm{i}k)$ *satisfy the inequality* $\mathrm{Re}(\lambda) \leq \alpha$*, with* α *being a positive constant.*

3.2 Stability analysis via the energy decay

It is also possible to work out a stability condition by enforcing energy decay over time. We left-multiply (3.3) by the conjugate transpose of $\hat{\boldsymbol{u}}$, denoted $\hat{\boldsymbol{u}}^*$:

$$\hat{\boldsymbol{u}}^* \frac{\mathrm{d}\hat{\boldsymbol{u}}}{\mathrm{d}t} = \hat{\boldsymbol{u}}^* \hat{P} \hat{\boldsymbol{u}}, \tag{3.5}$$

then we take the conjugate transpose of (3.3):

$$\frac{\mathrm{d}\hat{\boldsymbol{u}}^*}{\mathrm{d}t} = \hat{\boldsymbol{u}}^* \hat{P}^*,$$

with $\hat{P}^*$ denoting the adjoint of the symbol $\hat{P}$. Next step is to right-multiply last equation by $\hat{\boldsymbol{u}}$:

$$\frac{\mathrm{d}\hat{\boldsymbol{u}}^*}{\mathrm{d}t} \hat{\boldsymbol{u}} = \hat{\boldsymbol{u}}^* \hat{P}^* \hat{\boldsymbol{u}}, \tag{3.6}$$

and then add together (3.5) and (3.6) to obtain:

$$\hat{\boldsymbol{u}}^* \frac{\mathrm{d}\hat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{\mathrm{d}\hat{\boldsymbol{u}}^*}{\mathrm{d}t} \hat{\boldsymbol{u}} = \hat{\boldsymbol{u}}^* \hat{P} \hat{\boldsymbol{u}} + \hat{\boldsymbol{u}}^* \hat{P}^* \hat{\boldsymbol{u}}.$$

We note that the left-hand side of the last equation is indeed the time rate of change of an energy:

$$\hat{\boldsymbol{u}}^* \frac{\mathrm{d}\hat{\boldsymbol{u}}}{\mathrm{d}t} + \frac{\mathrm{d}\hat{\boldsymbol{u}}^*}{\mathrm{d}t} \hat{\boldsymbol{u}} = \frac{\mathrm{d}}{\mathrm{d}t}(\hat{\boldsymbol{u}}^* \hat{\boldsymbol{u}}) = \frac{\mathrm{d}}{\mathrm{d}t} \|\hat{\boldsymbol{u}}\|^2,$$

so that:

$$\frac{\mathrm{d}}{\mathrm{d}t} \|\hat{\boldsymbol{u}}\|^2 = \hat{\boldsymbol{u}}^* (\hat{P} + \hat{P}^*) \hat{\boldsymbol{u}}.$$

Since we want the energy to decay over time, we finally have the condition:

$$\hat{P} + \hat{P}^* < 0.$$

This proves the following theorem, in the special case $\alpha = 0$.

Theorem 3.2. *The initial value problem is well-posed if there is a constant α such that, for all k,*

$$\hat{P}(\mathrm{i}k) + \hat{P}^*(\mathrm{i}k) \leq 2\alpha I.$$

We note that in our case this last condition will be necessary but not sufficient, because we *assume* that the solution is periodic, but in general it is not. To put it in slightly different words, the above condition is useful to get a sense about what the parameters do, but it does not provide us with a complete picture, because we have assumed periodicity in space.

After this short detour, let's now go back to the differential operator for our problem:

$$P = -\begin{bmatrix} A_1\left(\dfrac{\partial}{\partial x_1} + \sigma_1\lambda_0\right) + A_2\left(\dfrac{\partial}{\partial x_2} + \sigma_2\tilde{\lambda}_0\right) & A_1\sigma_1 & A_2\sigma_2 \\ I\left(\dfrac{\partial}{\partial x_1} + \lambda_0(\alpha_0+\sigma_1) - \lambda_1\dfrac{\partial}{\partial x_2}\right) & I\left(\alpha_1\dfrac{\partial}{\partial x_2} + \alpha_0 + \sigma_1\right) & O \\ I\left(\dfrac{\partial}{\partial x_2} + \tilde{\lambda}_0(\tilde{\alpha}_0+\sigma_2) - \tilde{\lambda}_1\dfrac{\partial}{\partial x_1}\right) & O & I\left(\tilde{\alpha}_1\dfrac{\partial}{\partial x_1} + \tilde{\alpha}_0 + \sigma_2\right) \end{bmatrix},$$

and compute the symbol $\hat{P}$:

$$\hat{P} = -\begin{bmatrix} A_1(\mathrm{i}k_1 + \sigma_1\lambda_0) + A_2(\mathrm{i}k_2 + \sigma_2\tilde{\lambda}_0) & A_1\sigma_1 & A_2\sigma_2 \\ (\mathrm{i}k_1 + \lambda_0(\alpha_0 + \sigma_1) - \mathrm{i}\lambda_1 k_2)\, I & (\mathrm{i}\alpha_1 k_2 + \alpha_0 + \sigma_1)\, I & O \\ (\mathrm{i}k_2 + \tilde{\lambda}_0(\tilde{\alpha}_0 + \sigma_2) - \mathrm{i}\tilde{\lambda}_1 k_1)\, I & O & (\mathrm{i}\tilde{\alpha}_1 k_1 + \tilde{\alpha}_0 + \sigma_2)\, I \end{bmatrix}.$$

By enforcing the condition $\hat{P} + \hat{P}^* < 0$ the following inequalities can be obtained:

$$\begin{aligned} -2\lambda_0\sigma_1 &< 0, \\ -2\tilde{\lambda}_0\sigma_2 &< 0, \\ -\lambda_0(\alpha_0 + \sigma_1) + \mathrm{i}(k_2\lambda_1 - k_1) &< 0, \\ -\sigma_1 &< 0, \\ -\tilde{\lambda}_0(\tilde{\alpha}_0 + \sigma_2) + \mathrm{i}(k_1\tilde{\lambda}_1 - k_2) &< 0, \\ -\sigma_2 &< 0, \\ -2\sqrt{2}\lambda_0\sigma_1 &< 0, \\ -\sqrt{\sigma_1} &< 0, \\ -2\sqrt{2}\tilde{\lambda}_0\sigma_2 &< 0, \\ -\sqrt{\sigma_2} &< 0, \\ -\lambda_0(\alpha_0 + \sigma_1) + \mathrm{i}(k_1 - k_2\lambda_1) &< 0, \\ -2(\alpha_0 + \sigma_1) &< 0, \\ -\tilde{\lambda}_0(\tilde{\alpha}_0 + \sigma_2) + \mathrm{i}(k_2 - k_1\tilde{\lambda}_1) &< 0, \\ -2(\tilde{\alpha}_0 + \sigma_2) &< 0. \end{aligned}$$

Since $\sigma_1, \sigma_2 \geq 0$, the conditions on the parameters are:

$$\begin{aligned} \lambda_0 &> 0, \\ \tilde{\lambda}_0 &> 0, \\ \alpha_0 &> -\sigma_1, \\ \tilde{\alpha}_0 &> -\sigma_2. \end{aligned}$$

We note that the parameters α_1 and $\tilde{\alpha}_1$ disappear when we take $\hat{P} + \hat{P}^*$. Moreover, the parameters λ_1 and $\tilde{\lambda}_1$ are involved in the imaginary parts in which k_1 and k_2 do appear; this means that in principle they can take any value.

3.3 Stability analysis via continued fractions

Appelö et al. [2] proposed another way to study stability, without using the symbol of the differential operator. They studied the sign of the eigenvalues of the symbol $\hat{P}$ by means of the following theorem[1] by Frank [8], which can be found also in Marden [22].

Theorem 3.3 (Evelyn Frank, 1946)**.** *Consider any polynomial $q(z)$ of degree n. Let D be a real number and define the polynomials Q_0 and Q_1 with real coefficients by*

$$q(\mathrm{i}D) \equiv \mathrm{i}^n[Q_0(D) + \mathrm{i}Q_1(D)].$$

Then there is a continued fraction

$$\frac{Q_1(D)}{Q_0(D)} = \cfrac{1}{c_1 D + d_1 - \cfrac{1}{c_2 D + d_2 - \cfrac{1}{c_3 D + d_3 - \cdots - \cfrac{1}{c_{n_r} D + d_{n_r}}}}}$$

with $c_j \neq 0$ and $n_r \leq n$. The number of roots of $q(z)$ with positive (negative) real part equals the number of positive (negative) c_j. Moreover, there are $n - n_r$ roots on the imaginary axis.

There are several important points to notice about this theorem. The first one is that we must be able to write any polynomial $q(z)$ in such a way that we can read off the polynomials of real variable Q_0 and Q_1. This rewriting can *always* (and easily) be achieved. Secondly, we must be able to write the rational function Q_1/Q_0 in a continued fraction form. Also this can *always* be achieved, since there is an algorithm that makes recursive use of the Euclidean division between polynomials, no matter how complicated the rational function of departure, and returns the continued fraction expansion. Nonetheless, the calculations are far more tedious. Notice also that the total number of coefficients c_j appearing in the above expression may be less than the polynomial degree ($n_r \leq n$). Finally, we note that the theorem is also telling us that $n - n_r$ roots lie on the imaginary axis, namely those that have zero real part.

If we apply the above theorem to the characteristic polynomial of the symbol $\hat{P}$, we can determine the sign of its eigenvalues, which is exactly the information we need in order to apply the Petrovskii condition. This implies that *all the coefficients c_j in Theorem 3.3 must be defined and negative.*

Characteristic polynomial of $\hat{P}$. The characteristic polynomial $p(z)$ of the symbol $\hat{P}$ factorizes as

$$p(z) = z^2 \left(z + \tilde{\alpha}_0 + \mathrm{i}k_1\tilde{\alpha}_1\right)^6 \left(z + \alpha_0 + \mathrm{i}k_2\alpha_1 + \sigma_1\right)^2 \mu_4(z)\, \nu_4(z),$$

[1] This technique in fact turns out to be very similar to the Routh–Hurwitz stability criterion, often used in control theory. We point out this connection in Appendix D.

where $\mu_4(z)$ and $\nu_4(z)$ are two fourth degree polynomials:

$$\begin{aligned}\mu_4(z) = &\left(z^2 + k_1^2 + k_2^2\right)\left(z + \alpha_0 + \mathrm{i}k_2\alpha_1\right)^2 + 2\left(z + \alpha_0 + \mathrm{i}k_2\alpha_1\right) \times \\ &\times \left(k_2^2 + z\left(z - \mathrm{i}k_1\lambda_0\right) + k_1k_2\left(\alpha_1\lambda_0 + \lambda_1\right)\right)\sigma_1 + \\ &+ \left(-z^2\left(-1 + \lambda_0^2\right) - 2\mathrm{i}zk_2\lambda_0\left(\alpha_1\lambda_0 + \lambda_1\right) + k_2^2\left(1 + \left(\alpha_1\lambda_0 + \lambda_1\right)^2\right)\right)\sigma_1^2,\end{aligned}$$

$$\begin{aligned}\nu_4(z) = &\left(z^2 + 3k_1^2 + 3k_2^2\right)\left(z + \alpha_0 + \mathrm{i}k_2\alpha_1\right)^2 + 2\left(z + \alpha_0 + \mathrm{i}k_2\alpha_1\right) \times \\ &\times \left(z\left(z - 3\mathrm{i}k_1\lambda_0\right) + 3k_2\left(k_2 + k_1\left(\alpha_1\lambda_0 + \lambda_1\right)\right)\right)\sigma_1 + \\ &+ \left(z^2\left(1 - 3\lambda_0^2\right) - 6\mathrm{i}zk_2\lambda_0\left(\alpha_1\lambda_0 + \lambda_1\right) + 3k_2^2\left(1 + \left(\alpha_1\lambda_0 + \lambda_1\right)^2\right)\right)\sigma_1^2.\end{aligned}$$

It has to be noticed that the parameters $\tilde{\lambda}_0$ and $\tilde{\lambda}_1$ do not appear in the characteristic polynomial, which is a signal that they can take any value.

From the expression of its characteristic polynomial it is clear that $\hat{P}$ has two zero eigenvalues. Then there is six times the eigenvalue $z = -\tilde{\alpha}_0 - \mathrm{i}k_1\tilde{\alpha}_1$, which implies the condition

$$\tilde{\alpha}_0 > 0,$$

and twice the eigenvalue $z = -\alpha_0 - \mathrm{i}k_2\alpha_1 - \sigma_1$, which implies

$$\alpha_0 > -\sigma_1,$$

These two conditions agree with those found in Section 3.2.

The polynomials $\mu_4(z)$ and $\nu_4(z)$ are fourth degree polynomials, so, in principle, they admit an algebraic solution in closed-form. However, the closed-form expression of the solution is too long to allow analysis.

In what follows we apply Theorem 3.3 to $\mu_4(z)$ and $\nu_4(z)$ separately to work out their respective continued fraction expansions as the one showed in Theorem 3.3.

3.3.1 Application of Theorem 3.3 to $\mu_4(z)$

The first coefficient in the continued fraction expansion of $\mu_4(z)$ turns out to be:

$$c_1 = -\frac{1}{2(\alpha_0 + \sigma_1)},$$

which is defined if $\alpha_0 \neq -\sigma_1$, and it is negative if

$$\alpha_0 > -\sigma_1,$$

which agrees with the condition found above.

The second coefficient in the expansion is

$$\begin{aligned}c_2 = -2(\alpha_0 + \sigma_1)^3 / \big[&\alpha_0^4 + \alpha_0\left(k_1^2 + 4\alpha_0^2 - k_1k_2(2\alpha_1\lambda_0 + \lambda_1)\right)\sigma_1 + \\ &- \left(\alpha_0^2(-6 + \lambda_0^2) + k_1^2(-1 + \lambda_0^2) + k_1k_2(2\alpha_1\lambda_0 + \lambda_1)\right)\sigma_1^2 + \\ &- 2\alpha_0(-2 + \lambda_0^2)\,\sigma_1^3 - (-1 + \lambda_0^2)\,\sigma_1^4\big].\end{aligned}$$

Case $\sigma_1 \to 0$. To simplify the analysis, we consider the limit case in which we are approaching the PML interface, that is equivalent to $\sigma_1 \to 0$. In this case we drop the higher order terms in σ_1, and c_2 becomes

$$c_2 = -\frac{2(\alpha_0 + \sigma_1)^3}{\alpha_0^4 + \alpha_0 \left(k_1^2 + 4\alpha_0^2 - k_1 k_2 (2\alpha_1 \lambda_0 + \lambda_1)\right) \sigma_1}.$$

Then the questions remain the same: *is c_2 defined?* and, if it is defined, *is it negative?* To answer both these questions, we have to check when the denominator of c_2

$$f(k_1, k_2) = \alpha_0^4 + \alpha_0 \left(k_1^2 + 4\alpha_0^2 - k_1 k_2 (2\alpha_1 \lambda_0 + \lambda_1)\right) \sigma_1 \tag{3.7}$$

is positive.

To gain insight into the sign of the denominator we plot the surface $f(k_1, k_2)$ for a given set of the parameters. Figure 3.1 shows the surface of $f(k_1, k_2)$ together with the horizontal plane $z = 0$, in the domain $(k_1, k_2) \in [-1000, +1000]^2$.

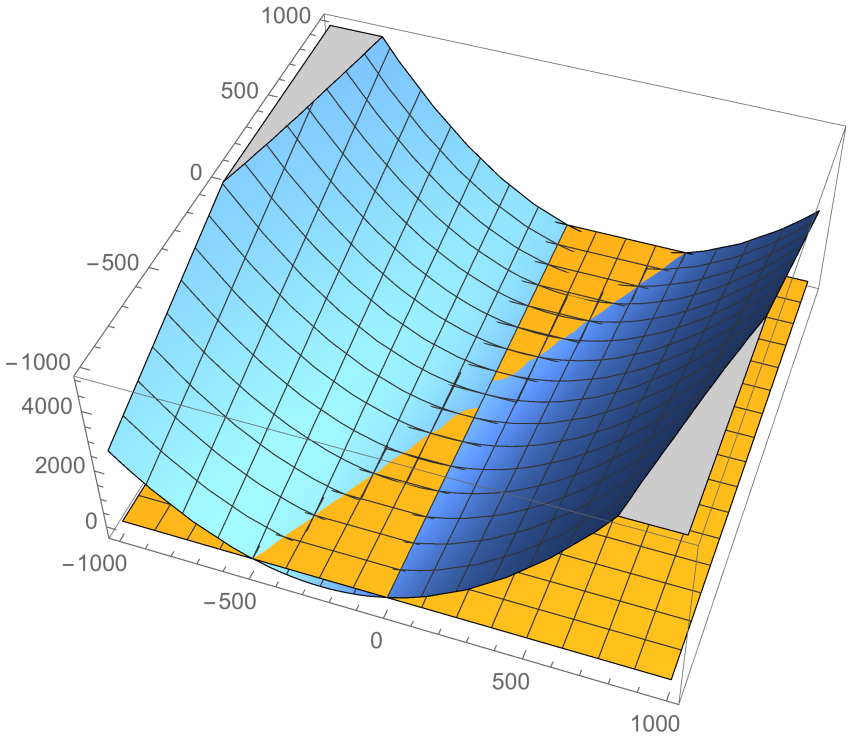


Figure 3.1: Plot of $f(k_1, k_2)$ for some set of the parameters, with $(2\alpha_1 \lambda_0 + \lambda_1) \neq 0$.

Since $f(k_1, k_2)$ negative implies c_2 positive, the regions of instability are those where $f(k_1, k_2)$ is negative. It is evident from Figure 3.1 that there are regions where $f(k_1, k_2)$ is negative, and the boundaries of such regions are given by the intersection between the surface $f(k_1, k_2)$ and the horizontal plane $z = 0$. Figure 3.2 shows the two branches of the equation $f(k_1, k_2) = 0$. The triangular regions are regions of instability, as in these regions c_2 is positive.

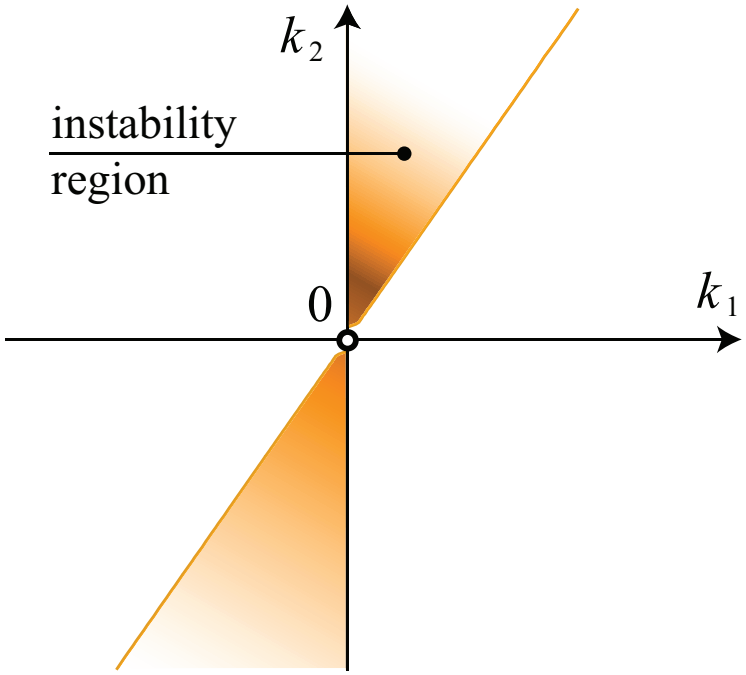


Figure 3.2: The instability region implied by c_2.

In what follows, we seek an analytical expression for the boundaries of the instability regions. Starting from $f(k_1, k_2) > 0$, we work out k_2 as a function of k_1

$$\alpha_0^3 + \left(k_1^2 + 4\alpha_0^2 - k_1 k_2 (2\alpha_1\lambda_0 + \lambda_1)\right)\sigma_1 > 0,$$

$$\alpha_0^3 + k_1^2\sigma_1 + 4\alpha_0^2\sigma_1 - k_1 k_2 (2\alpha_1\lambda_0 + \lambda_1)\sigma_1 > 0,$$

$$\alpha_0^3 + k_1^2\sigma_1 + 4\alpha_0^2\sigma_1 > k_1 k_2 (2\alpha_1\lambda_0 + \lambda_1)\sigma_1.$$

For $k_1 = 0$ one has

$$\alpha_0^3 + 4\alpha_0^2\sigma_1 > 0,$$

which is always guaranteed. Instead, for $k_1 \neq 0$ one obtains

$$\begin{cases} k_2 < \dfrac{\alpha_0^3 + (4\alpha_0^2 + k_1^2)\,\sigma_1}{(2\alpha_1\lambda_0 + \lambda_1)\,k_1\sigma_1} & \text{if } k_1 > 0, \\ k_2 > \dfrac{\alpha_0^3 + (4\alpha_0^2 + k_1^2)\,\sigma_1}{(2\alpha_1\lambda_0 + \lambda_1)\,k_1\sigma_1} & \text{if } k_1 < 0. \end{cases} \tag{3.8}$$

Since we would like to have no condition on k_2, i.e. it should be allowed to take any value, the only possibility is that the expression on the right-hand side of the last inequalities is unbounded

$$\frac{\alpha_0^3 + (4\alpha_0^2 + k_1^2)\,\sigma_1}{(2\alpha_1\lambda_0 + \lambda_1)\,k_1\sigma_1} \to \infty, \quad \forall k_1 \neq 0,$$

and since $k_1\sigma_1 \neq 0$, the only possibility is that

$$(2\alpha_1\lambda_0 + \lambda_1) \to 0, \tag{3.9}$$

which means that either $\lambda_0 = \lambda_1 = 0$, or $\alpha_1 = \lambda_1 = 0$ or $\lambda_0 = -\lambda_1/2\alpha_1$. If we assume that all the parameters are positive, then the third condition has to be discarded. Moreover, all the numerical simulations have shown that in practice λ_0 has to stay zero, so that we are left only with the first condition

$$\lambda_0 = \lambda_1 = 0. \tag{3.10}$$

In general, we can observe that the presence of the instability region is associated with the presence of the mixed term in $k_1 k_2$ in (3.7). If in some way the coefficient of this mixed term is zero, then $f(k_1, k_2)$ is always positive, it never intersects the horizontal plane $z = 0$, and hence c_2 is negative. Figure 3.3 illustrates this last case.

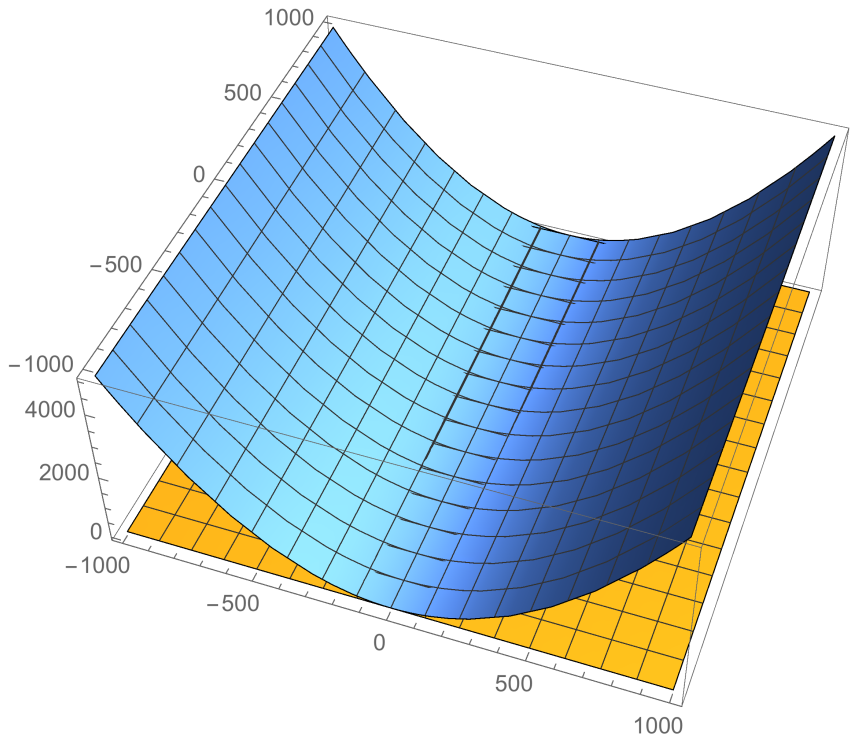


Figure 3.3: Plot of $f(k_1, k_2)$ for some set of the parameters, with $(2\alpha_1\lambda_0 + \lambda_1) = 0$.

It can be seen that, by following the same procedure for $\nu_4(z)$, one can find the same results. The coefficients c_3 and c_4 of the continued fraction expansions, however, have more complicated expressions that do not allow analysis.

Furthermore, we recall that we are only looking at the limit in which $\sigma_1 \to 0$, so the above conditions may not be giving us the complete picture. However, the analysis is confirmed by the fact that all the numerical simulations performed with $\lambda_0 \neq 0, \lambda_1 \neq 0, \alpha_0 \neq 0, \alpha_1 \neq 0$ show that in these cases the BGK+PML model is unstable. On the contrary, the simulations performed with $\lambda_0 = 0, \lambda_1 = 0, \alpha_0 \neq 0, \alpha_1 \neq 0$ show that in these cases the BGK+PML model is stable, but then there is an accuracy issue to tackle. In the next chapter we are going to further develop these aspects.

4

Sensitivity analysis

> *The theory of integration formulas with polynomial degrees of accuracy is closely connected to the theories of polynomial interpolation and orthogonal polynomials. Both fields are considerably more complex when dealing with several variables than when dealing with just one.*
>
> – Krommer and Ueberhuber

Now that we have established bounds on the parameters appearing in the BGK+PML model, we need to assess how they impact the outcome of the simulations.

In the literature we can find many methods to sample a parameter space and perform a sensitivity analysis. A good review is given in [1]. There the author points out how models with many parameters often behave as if they really depend on only a few, which impact the value of the target outcome. Besides yielding accurate results, a good sampling method should also minimize the number of times a model has to be run.

The most straightforward way to explore a parameter space would be to generate a simple random sample of the parameters in some given intervals of variation. Unfortunately, random sampling has a convergence rate of $1/\sqrt{d}$, because of the law of large numbers. In principle, a random sampling is correct, but if one has many parameters, it proves to be inefficient and computationally expensive. In the most general case that will be considered here, the parameter space will be 4-dimensional. Hence we need to find a better way to systematically explore the parameter space.

The tool that we will use to gain insights into the role of our parameters is the Analysis of Variance (ANOVA) expansion of multivariate functions and the connected concept of Total Sensitivity Indices (TSIs) [5, 10, 24]. ANOVA expansions turn out to be very useful when one wants to study functionals of solutions to nonlinear partial differential equations. But before going deeper into these concepts, we define our functional of interest.

4.1 Definition of the error functional

We have to define an outcome of the solution on which to focus our attention. The preferred outcome will be a functional of the solution to our BGK+PML model. In particular, we can choose it to be the maximum over time of the L^2-norm of the

error in the density a_0 between the BGK+PML and the plain BGK, calculated along a line close to the PML, and normalized with respect to the L^2-norm of the initial condition of the density a_0 on that same line for the plain BGK. Synthetically:

$$g_1 = \frac{\max_{t\in[0,T]} \left\{ \left[\int_0^{L_y} \left(a_0^{\text{BGK+PML}}(x^*, y, t) - a_0^{\text{BGK}}(x^*, y, t) \right)^2 \, \mathrm{d}y \right]^{1/2} \right\}}{\left[\int_0^{L_y} \left(a_0^{\text{BGK}}(x^*, y, t = 0) \right)^2 \, \mathrm{d}y \right]^{1/2}}, \tag{4.1}$$

where T is the total simulated time, x^* is the abscissa at which the reference line is located, while $a_0^{\text{BGK+PML}}$ and a_0^{BGK} are the densities with and without the PML, respectively. In the most general case that we will consider, this error functional will depend on four parameters, namely α_0, α_1, the PML exponent β and the PML thickness L. In Section 4.7.1 we will also make use of other functionals, but until then we will stick to (4.1).

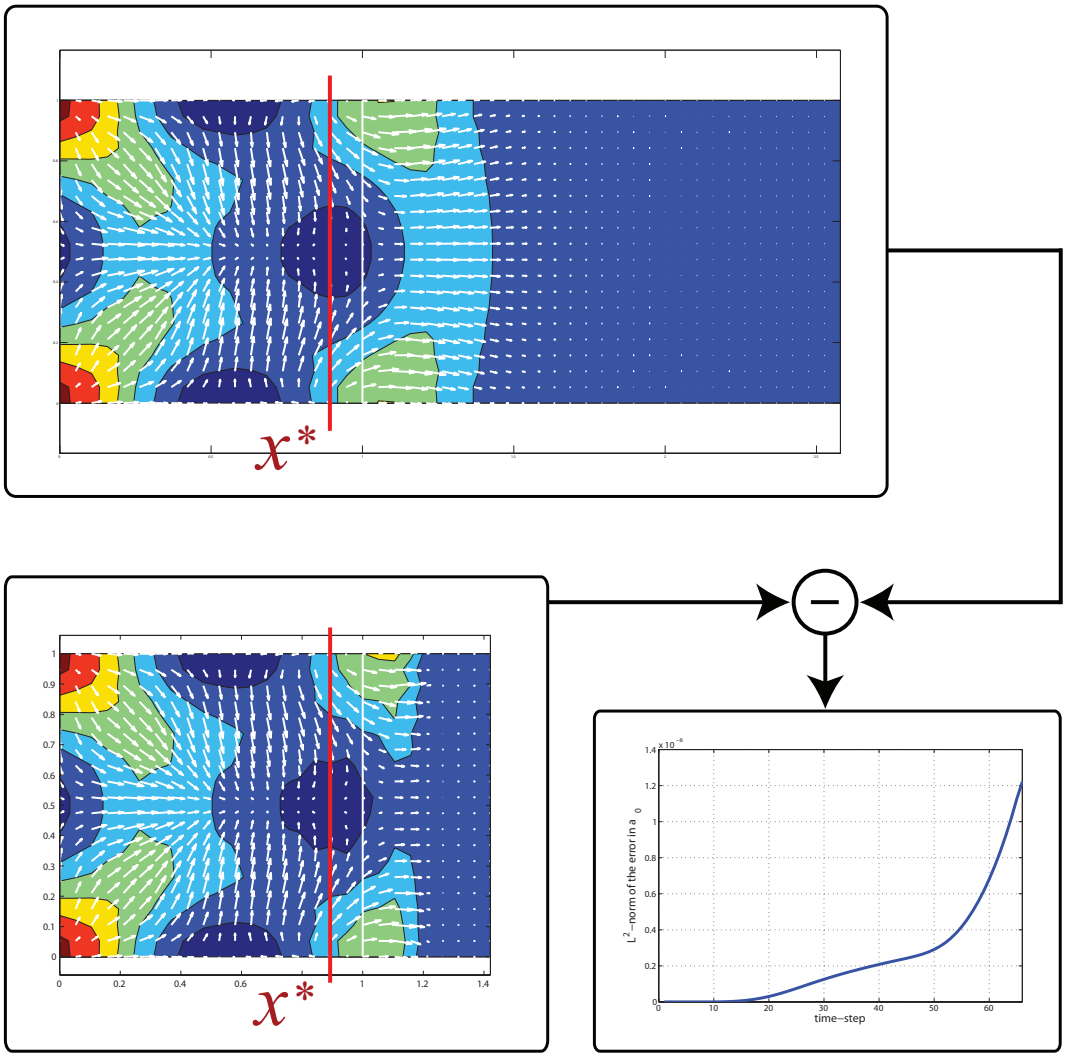


Figure 4.1: Illustration of the construction of expression (4.2).

4.2 Bounds on the parameters

4.2.1 λ_0 and λ_1 have to stay zero

In Section 3.3.1 we have seen from an analytical point of view that in order to guarantee stability we must freeze the parameters λ_0 and λ_1 to zero. Here we are going to show that the numerical simulations confirm these stability conditions. As in (4.1), we consider the quantity

$$\frac{\left[\int_0^{L_y} \left(a_0^{\mathrm{BGK+PML}}(x^*, y, t) - a_0^{\mathrm{BGK}}(x^*, y, t)\right)^2 \, \mathrm{d}y\right]^{1/2}}{\left[\int_0^{L_y} \left(a_0^{\mathrm{BGK}}(x^*, y, t = 0)\right)^2 \, \mathrm{d}y\right]^{1/2}}, \tag{4.2}$$

and plot its time evolution, by performing a numerical simulation of the BGK+PML model with the same data described in Section 2.7. Figure 4.1 qualitatively illustrates how expression (4.2) is computed.

From Figure 4.2 we observe that the error is very small throughout the simulation, remaining in the order of 10^{-6}, and also the visual outcomes (not reported here) appear to be reasonable from the physical point of view. Now we redo the

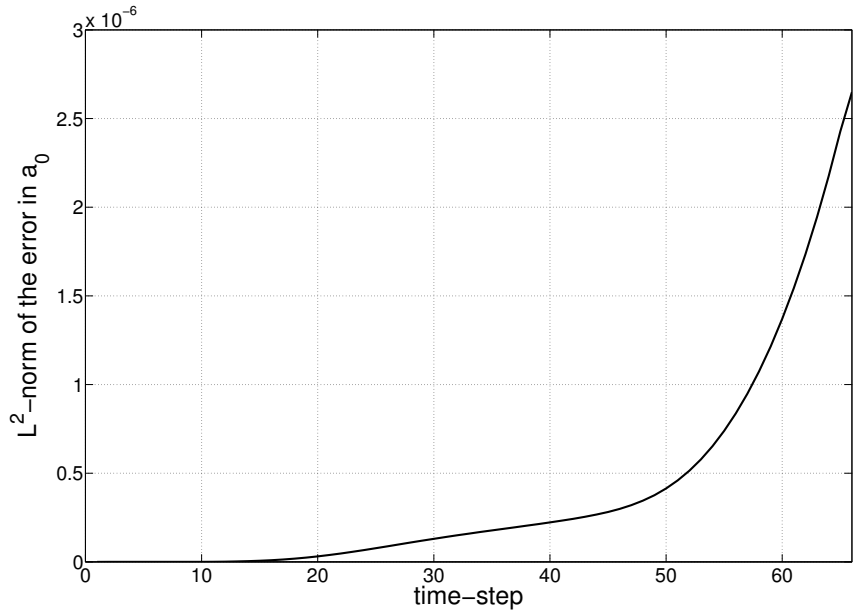


Figure 4.2: Time evolution of the L^2-norm of the error in a_0 on a vertical line close to the PML. Reasonable results.

same simulation but setting $\lambda_0 = 0.05$. The time evolution of the error is reported in Figure 4.3, which shows that now the error is four order of magnitudes larger than the one in Figure 4.2. It can also be seen from the visual outcomes (not reported here) that the accuracy of the method in reproducing the physical behaviour is irreversibly compromised. From all the numerical simulations performed it appears that we must set λ_0 and λ_1 to zero to address the stability issues and to ensure an accurate reproduction of the physical behaviour. This is in agreement with the stability analysis carried out in Section]3.3.1. We finally note that, even if we freeze λ_0 to zero, the parameter α_0 still appears in the equations (2.5).

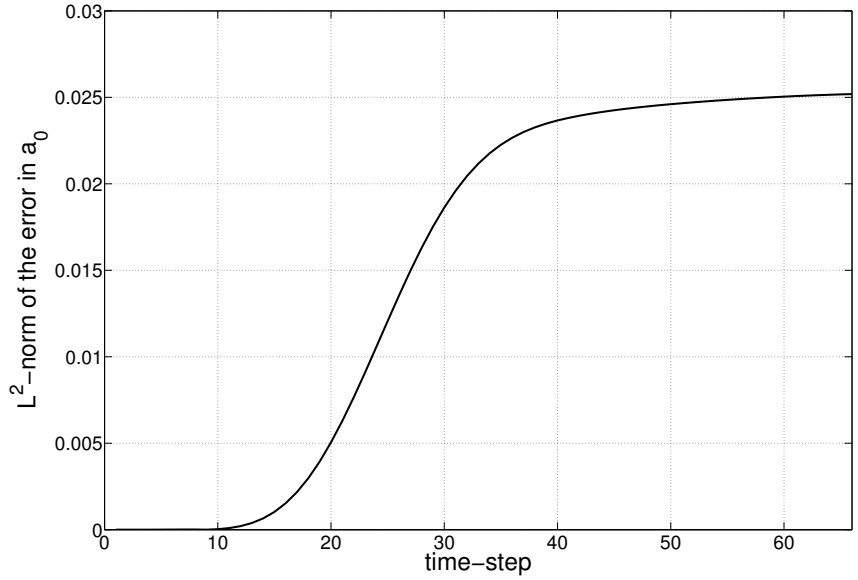


Figure 4.3: Time evolution of the L^2-norm of the error in a_0 on a vertical line close to the PML. Unreasonable results.

4.2.2 Bounds on the PML thickness

Later we will consider also the PML thickness L as a parameter to be analysed, so we proceed to establish some appropriate lower bound on L before diving into the ANOVA expansion. Figure 4.4 shows the time evolution of the L^2-norm of the error in a_0 on a vertical line close to the PML, for several PML thicknesses, $L = \{0.10, 0.15, 0.20, 0.25\}$. By looking at Figure 4.4 we can conclude that a reasonable lower bound on the PML thickness L can be given by $L = 0.25$.

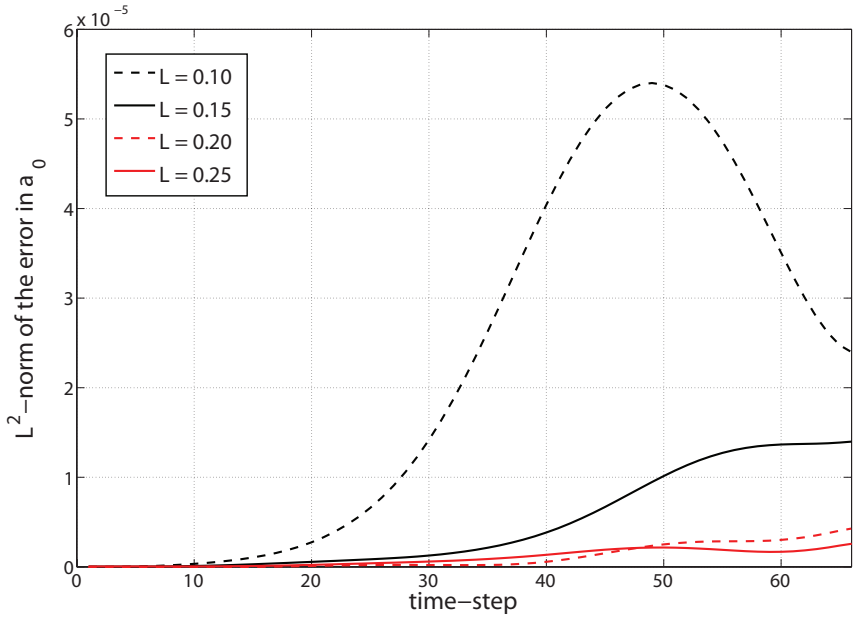


Figure 4.4: Time evolution of the L^2-norm of the error in a_0 on a vertical line close to the PML, for several PML thicknesses.

4.3 ANOVA expansion of multivariate functions

Let $\mathcal{P} = \{1, 2, \ldots, p\}$ be the set of coordinate indices of a p-dimensional function, and let $\boldsymbol{\alpha} = (\alpha^1, \alpha^2, \cdots, \alpha^p) \in \mathbb{R}^p$ be a p-dimensional vector. Let $\mathcal{T} \subseteq \mathcal{P}$ be a subset of $\mathcal{P}$, and let t denote the cardinality of $\mathcal{T}$, i.e. the number of elements in $\mathcal{T}$. We denote by $\boldsymbol{\alpha}_\mathcal{T} \in \mathbb{R}^t$ the t-dimensional vector that contains the components of $\boldsymbol{\alpha} \in \mathbb{R}^p$ indexed by $\mathcal{T}$. Furthermore, we denote by A^p the p-dimensional unit hypercube $[0, 1]^p$, and A^t the t-dimensional unit hypercube which is the projection of the p-dimensional unit[1] hypercube A^p onto the coordinates indexed by $\mathcal{T}$. Then any p-dimensional function $g \in L^2(A^p)$ can be written as the *ANOVA expansion* [5, 10, 24]:

$$g(\boldsymbol{\alpha}) = g_0 + \sum_{\mathcal{T} \subseteq \mathcal{P}} g_\mathcal{T}(\boldsymbol{\alpha}_\mathcal{T}), \tag{4.3}$$

where the terms in the expansion are calculated recursively through

$$g_\mathcal{T}(\boldsymbol{\alpha}_\mathcal{T}) = \int_{A^{p-t}} g_\mathcal{T}(\boldsymbol{\alpha}_\mathcal{T}) \, \mathrm{d}\boldsymbol{\alpha}_{\mathcal{P}\setminus\mathcal{T}} - \sum_{\mathcal{W} \subset \mathcal{T}} g_\mathcal{W}(\boldsymbol{\alpha}_\mathcal{W}) - g_0, \tag{4.4}$$

starting with the zero-th order term (which is just a constant):

$$g_0 = \int_{A^p} g(\boldsymbol{\alpha}) \, \mathrm{d}\boldsymbol{\alpha},$$

and where, by convention,

$$\int_{A^0} g(\boldsymbol{\alpha}) \, \mathrm{d}\boldsymbol{\alpha}_\emptyset = g(\boldsymbol{\alpha}).$$

Each term $g_\mathcal{T}(\boldsymbol{\alpha}_\mathcal{T})$ in the ANOVA expansion is, in general, a nonlinear function of its t arguments, and it is the *unique* term in the expansion that depends on the t variables indexed by $\mathcal{T}$. In other words, the term $g_\mathcal{T}(\boldsymbol{\alpha}_\mathcal{T})$ describes the effect within $g(\boldsymbol{\alpha})$ when those t arguments are simultaneously taken into account.

We emphasize that $\mathrm{d}\boldsymbol{\alpha}_{\mathcal{P}\setminus\mathcal{T}}$ in (4.4) indicates integration over all those coordinate indices *not* included in $\mathcal{T}$, and that the sum is carried out over *strict* subsets $\mathcal{W}$ of $\mathcal{T}$. The operation (4.4) can actually be regarded as a *projection*, since the resulting function will depend only on the coordinate indices contained in $\mathcal{T}$. The total number of terms in the ANOVA expansion is 2^p.

We point out that the ANOVA expansion is *exact* and contains a *finite* number of terms, even though we can *truncate* it to obtain a good approximation to $g(\boldsymbol{\alpha})$, *having less terms* than the full expansion. Then the natural question arises about what is meant by good approximation, and trying to answer this question leads us to the concept of *effective dimension of multivariate functions* (see sections 4.3.2 and 4.3.3).

We hope to make things clearer by writing down explicitly the expressions to calculate the first few terms in the ANOVA expansion. We define the *order* of a term $g_\mathcal{T}(\boldsymbol{\alpha}_\mathcal{T})$ appearing in the ANOVA expansion (4.3) as the cardinality t of the

[1] We note that the keyword here is *unit*. We will come back to this later.

corresponding set $\mathcal{T}$. So the case $t = 1$ generates the *first order terms*, or *univariate functions*, given by

$$g_i(\alpha_i) = \int_{A^{p-1}} g(\alpha)\, \mathrm{d}\boldsymbol{\alpha}' - g_0, \qquad i = 1, 2, \ldots, p,$$

where $\mathrm{d}\boldsymbol{\alpha}'$ indicates integration over all coordinates except α_i.

The case $t = 2$ generates the *second order terms*, or *bivariate functions*, given by

$$g_{ij}(\alpha_i, \alpha_j) = \int_{A^{p-2}} g(\boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}'' - g_i(\alpha_i) - g_j(\alpha_j) - g_0, \qquad i < j, \quad i, j = 1, 2, \ldots, p,$$

where $\mathrm{d}\boldsymbol{\alpha}''$ indicates integration over all coordinates except α_i and α_j.

The case $t = 3$ generates the *third order terms*, or *trivariate functions*, given by

$$\begin{aligned} g_{ijk}(\alpha_i, \alpha_j, \alpha_k) = \int_{A^{p-3}} g(\boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}''' - g_{ij}(\alpha_i, \alpha_j) - g_{ik}(\alpha_i, \alpha_k) - g_{jk}(\alpha_j, \alpha_k) \\ - g_i(\alpha_i) - g_j(\alpha_j) - g_k(\alpha_k) - g_0, \qquad i < j < k, \quad i, j, k = 1, 2, \ldots, p, \end{aligned}$$

where $\mathrm{d}\boldsymbol{\alpha}'''$ indicates integration over all coordinates except α_i, α_j and α_k, and so on. Note that, as we go to higher order, i.e. as t increases, the dimensionality of the integrals that we need to compute to construct the expansion decreases. Moreover, the total number of t-th order terms is given by the binomial coefficient

$$\binom{p}{t} = \frac{p!}{t!\,(p-t)!}.$$

The ANOVA expansion of $g(\boldsymbol{\alpha})$ is finally written as

$$g(\boldsymbol{\alpha}) = g_0 + \sum_{i}^{p} g_i + \sum_{i,j}^{\binom{p}{2}} g_{ij} + \sum_{i,j,k}^{\binom{p}{3}} g_{ijk} + \cdots .$$

From a computational point of view it is apparent that the bottleneck is given by the evaluation of the many multidimensional integrals needed to construct the expansion. This aspect should not be underestimated because it might be a hindrance to the efficiency of the algorithm. We will discuss multivariate integration techniques later in Section 4.4.

4.3.1 Properties of the ANOVA expansion

In this section we provide a list of the most important properties of the ANOVA expansion, although some of them have already been mentioned in the previous section. A full list can be found in [24].

- The ANOVA expansion of a general p-dimensional function $g \in L^2(A^p)$ is *exact* and *finite*, and contains a total number of 2^p terms;
- the *zero-th order term* g_0 in (4.3) is an integral average of g over the entire parameter space A^p, and it is a *constant*;

- the generic term $g_{\mathcal{T}}(\boldsymbol{\alpha}_{\mathcal{T}})$ is a function *only* of the coordinates indexed by $\mathcal{T}$;
- *the terms in the ANOVA expansion are mutually orthogonal*, namely

$$\int_{A^p} g_{\mathcal{S}}(\boldsymbol{\alpha}_{\mathcal{S}})\, g_{\mathcal{T}}(\boldsymbol{\alpha}_{\mathcal{T}})\, \mathrm{d}\boldsymbol{\alpha} = 0,$$

  whenever $\mathcal{S}$ and $\mathcal{T}$ contain at least one different index. This holds also when $\mathcal{S}$ and $\mathcal{T}$ have the same cardinality. Note that when $g_{\mathcal{S}} = g_0$ we get the particular case

$$\int_{A^p} g_0\, g_{\mathcal{T}}(\boldsymbol{\alpha}_{\mathcal{T}})\, \mathrm{d}\boldsymbol{\alpha} = 0,$$

  which, since g_0 is constant, implies

$$\int_{A^p} g_{\mathcal{T}}(\boldsymbol{\alpha}_{\mathcal{T}})\, \mathrm{d}\boldsymbol{\alpha} = 0,$$

  meaning that *the terms $g_{\mathcal{T}}(\boldsymbol{\alpha}_{\mathcal{T}})$ in the ANOVA expansion have zero average.*
- each term $g_{\mathcal{T}}(\boldsymbol{\alpha}_{\mathcal{T}})$ in the expansion is a *projection* of $g(\boldsymbol{\alpha})$ onto a subspace of $L^2(A^p)$, with respect to the $L^2(A^p)$ inner product.

4.3.2 The truncated ANOVA expansion

Definition 4.1. *A truncated ANOVA expansion of order r is given by*

$$g(\boldsymbol{\alpha}; r) = g_0 + \sum_{\mathcal{T}\subseteq\mathcal{P},\ t\leq r} g_{\mathcal{T}}(\boldsymbol{\alpha}_{\mathcal{T}}). \tag{4.5}$$

The truncated ANOVA expansions are of great importance because it can often be observed that $g(\boldsymbol{\alpha}; r)$ with $r \ll p$ already yields a good approximation to $g(\boldsymbol{\alpha})$.

Some consequences of approximating well a multivariate function $g(\boldsymbol{\alpha})$ by a truncated ANOVA expansion $g(\boldsymbol{\alpha}; r)$ are presented in [5]. For instance, if $r \ll p$, then our $g(\boldsymbol{\alpha})$, which is a function of p arguments, can be well[2] described by a sum of terms each of which depends at most on r variables. This means that *the contributions provided by coordinate sets having more than r variables can be disregarded.* This leads us to the concept of effective dimension of a function.

4.3.3 The effective dimension of a function

As we have mentioned above, the ANOVA expansion is related to the concept of effective dimension of a multivariate function [5].

Definition 4.2. *The effective dimension of a multivariate function g in the superposition sense or, in short, the* superposition dimension*, is the smallest integer r such that*

$$\sum_{0<t\leq r} V_{\mathcal{T}}(g) \geq q\, V(g),$$

where $q > 0$ is called proportion *and it is typically chosen to be slightly less than 1; $q = 0.99$ is a common choice.*

[2] In the following section we will state more precisely what do we mean by *well*.

Here the terms $V_{\mathcal{T}}(g)$ and $V(g)$ are defined by

$$V_{\mathcal{T}}(g) = \int_{A^p} \left(g_{\mathcal{T}}(\boldsymbol{\alpha}_{\mathcal{T}})\right)^2 \, \mathrm{d}\boldsymbol{\alpha}, \qquad V(g) = \sum_{t>0} V_{\mathcal{T}}(g). \tag{4.6}$$

Note that $V_{\mathcal{T}}(g)$ is the integral average of the square of the terms appearing in the ANOVA expansion, and can be regarded as a variability of g over a given set $\mathcal{T}$.

Definition 4.3. *Given a function g and its approximation h, the* normalized approximation error *is defined by*

$$E(g,h) = \frac{1}{V(g)} \int_{A^p} \left(g(\boldsymbol{\alpha}) - h(\boldsymbol{\alpha})\right)^2 \, \mathrm{d}\boldsymbol{\alpha}.$$

We have the following remarkable theorem about the approximation property of the truncated ANOVA expansions.

Theorem 4.1. *Assume that $g(\boldsymbol{\alpha})$ has superposition dimension r in proportion q and let $g(\boldsymbol{\alpha}; r) = \sum_{0<t<r} g_{\mathcal{T}}(\boldsymbol{\alpha}_{\mathcal{T}})$ denote its truncated ANOVA expansion of order r. Then*

$$E\left(g(\boldsymbol{\alpha}), g(\boldsymbol{\alpha}; r)\right) \leq (1-q).$$

This theorem clarifies what we claimed in the previous section, i.e., if the superposition dimension is small ($r \ll p$), then $g(\boldsymbol{\alpha})$ can be well approximated by a truncated ANOVA expansion with only few terms. But what does the adverb *well* in that sentence mean? By the previous theorem, it means that the error between the ANOVA expansion and its truncation of order r is less than or equal to $(1-q)$.

It has been shown in many practical applications that truncated ANOVA expansions of order two can already yield very good approximations to the original function g [5, 10]. Usually the reality is that high-dimensional functions are not truly high-dimensional. Multidimensional functions that really depend on the connection between all the parameters are found quite seldom. In particular, usually the bivariate terms in the ANOVA expansion do still matter, but if we take into account even higher order terms, then we find that they make a small difference. The ANOVA expansion is very useful because identifies how much structure hides behind a multivariate function.

To illustrate that higher order terms always bring a minor contribution, we consider a Gaussian test function

$$f(\boldsymbol{x}) = \exp\left[-\textstyle\sum_{i=1}^{p} c_i^2 (x_i - \omega_i)^2\right], \tag{4.7}$$

where the coefficients $\boldsymbol{c} = (c_1, \ldots, c_p)$ and $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_p)$ are generated randomly.

We carry out the ANOVA expansion of a 4D Gaussian whose coefficients are

$$\boldsymbol{c} = (0.122, 0.3627, 0.908, 0.5756),$$

$$\boldsymbol{\omega} = (0.989, 0.567, 0.9898, 0.3223)$$

Figure 4.5 shows the L^2-norm of the error between the 4-dimensional Gaussian test function and its truncated ANOVA expansions as a function of the order of the truncated ANOVA expansion.

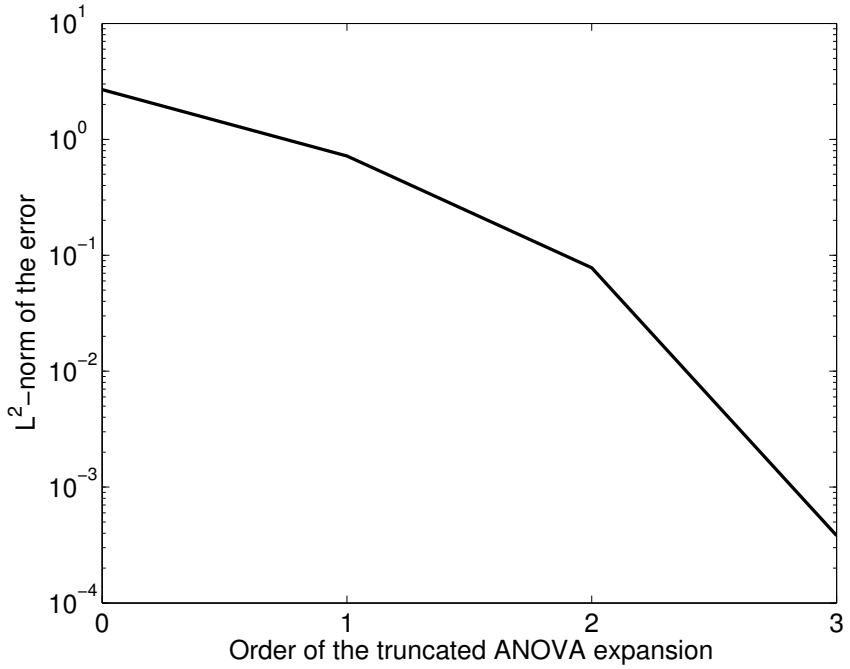


Figure 4.5: Convergence behaviour of the truncated ANOVA expansion of a 4D Gaussian test function as a function of the truncation order.

We can observe from Figure 4.5 that the contributions to the error become smaller and smaller as the truncation order of the ANOVA expansion increases.

4.3.4 Total Sensitivity Indices

The Total Sensitivity Index (TSI) of a parameter α_i measures the combined sensitivity of all terms that depend on α_i, $i = 1, \ldots, p$. We define the sensitivity measure

$$S_{\mathcal{T}} = \frac{V_{\mathcal{T}}}{V},$$

where $V_{\mathcal{T}}$ and V are defined according to (4.6). The following result holds:

$$\sum_{i \in \mathcal{T}} S_{\mathcal{T}} + \sum_{i \notin \mathcal{T}} S_{\mathcal{T}} = 1,$$

where the first term is a sum of the sensitivity measures $S_{\mathcal{T}}$ which contain the coordinate index i and the second term is a sum of those $S_{\mathcal{T}}$ that do not contain it. We call the first term in the above expression the TSI(i) of variable α_i. These TSIs give us a feeling of which parameters are most important, and represent the final goal of our computation of the ANOVA expansion. The multidimensional integrals required to construct the ANOVA expansion do not need to be computed very accurately, since from the TSIs we just want to get a sense of which parameters matter the most.

4.4 Multivariate numerical integration

We have seen that a central ingredient to compute the ANOVA expansion is to be able to evaluate in a reasonable way the multidimensional integrals appearing in (4.4). All the theory about the ANOVA expansion is based on the assumption that these multidimensional integrals can be evaluated *exactly*. In our case this is not possible since we do not have an analytic expression for g. When implementing, this implies that we have to resort to multivariate numerical integration. The efficiency and the accuracy of the integration methods adopted will affect the efficiency of the calculations and the accuracy of the ANOVA expansion.

There are several approaches to evaluate multidimensional integrals, for instance:

- the *Stroud cubature*, which is the simplest approach and gives the minimum amount of nodes to obtain a certain accuracy in high dimensions, but it cannot provide very high accuracy;
- *product rules*, that allow to extend the many known univariate integration formulas to higher dimensions. This approach allows to calculate the integrals accurately, but it quickly becomes computationally expensive, because the number of samples grows like n^p for a quadrature using n points in p dimensions. For instance, the Cartesian product with a Gaussian quadrature for 7 dimensions with 5 integration abscissas in each dimension needs $5^7 = 78125$ evaluations, which is already a huge amount.
- the Smolyak construction, which is a sparse grid integration method.

We note that our final goal is to compute the TSIs of the parameters in order to get a feeling of how they affect our output functional g. The evaluation of the multivariate integrals does not need to be very accurate, because we are not interested in the actual values of the TSIs, but in understanding how the parameters relate to each other and which one is more important.

In what follows we review the integration techniques that have been considered to compute the ANOVA expansion: the Stroud cubature formulas and the product rules.

4.4.1 Stroud cubature

The simplest approach to perform a multi-dimensional numerical integration is given by the so-called Stroud cubature points [25, 26], which can be used to calculate integrals of the type:

$$I[g] = \int_{[-1,1]^p} g(\boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}, \tag{4.8}$$

where g is our p-dimensional functional, $\boldsymbol{\alpha} = (\alpha^1, \alpha^2, \cdots, \alpha^p)$ are the parameters and $[-1,1]^p$ is the p-dimensional reference hypercube. The only difference between (4.8) and the integrals appearing in (4.4) is that the ANOVA expansion considers the p-dimensional unit hypercube $[0,1]^p$, while the Stroud cubature works for the hypercube $[-1,1]^p$. This suggests the use of a mapping in order to make the Stroud cubature available to the unit hypercube $[0,1]^p$ (see Section 4.4.2).

To be rigorous, we should write $g[u(\boldsymbol{\alpha})]$ instead of $g(\boldsymbol{\alpha})$, since g is actually a functional of the solution $u(\boldsymbol{\alpha})$ resulting from our simulation of the BGK model coupled with the PML. In the following, we will stick to this little abuse of notation, and simply write $g(\boldsymbol{\alpha})$.

Stroud cubature formula of degree 2

The Stroud cubature formula of degree 2 adopts $p+1$ equally weighted points (i.e., one more than the number of parameters we consider). The following theorem is of interest [25]:

Theorem 4.2. *A necessary and sufficient condition that $p+1$ equally weighted points form a numerical integration formula of degree 2 for a symmetric region or for a regular p-simplex is that these points form the vertices of a regular p-simplex whose centroid coincides with the centroid of the region and lie on the surface of a sphere of radius $r = \sqrt{p\, I_2/I_0}$, where*

$$I_0 = \int_R \mathrm{d}v, \qquad I_2 = \int_R x_1^2 \,\mathrm{d}v = \cdots = \int_R x_n^2 \,\mathrm{d}v.$$

Here I_0 is the hypervolume, and the weight of the points is $I_0/(p+1)$.

This cubature formula yields the following approximation to the integral in (4.8):

$$I[u] \simeq \frac{2^p}{p+1} \sum_{i=1}^{p+1} g(\boldsymbol{\alpha}_i),$$

where $\boldsymbol{\alpha}_i = (\alpha_i^1, \alpha_i^2, \cdots, \alpha_i^p)$, with $i = 1, 2, \ldots, p+1$, are the $p+1$ p-dimensional cubature points. The factor 2^p at the numerator of the fraction in front of the sum is actually the volume of the hypercube $[-1,1]^p$. The explicit point location is given by:

$$\alpha_i^{2r-1} = \sqrt{\frac{2}{3}} \cos\left(\frac{2r\,(i-1)\,\pi}{p+1}\right), \quad \alpha_i^{2r} = \sqrt{\frac{2}{3}} \sin\left(\frac{2r\,(i-1)\,\pi}{p+1}\right),$$
$$r = 1, 2, \ldots, [p/2],$$

where $[p/2]$ is the greatest integer not exceeding $p/2$, and if p is odd $\alpha_i^p = (-1)^{(i-1)}/\sqrt{3}$.

Stroud cubature formula of degree 3

The Stroud cubature formula of degree 3 makes use of $2p$ equally weighted points to approximate (4.8) in the following way:

$$I[u] \simeq \frac{2^{p-1}}{p} \sum_{i=1}^{2p} g(\boldsymbol{\alpha}_i),$$

where $\boldsymbol{\alpha}_i = (\alpha_i^1, \alpha_i^2, \cdots, \alpha_i^p)$, with $i = 1, 2, \ldots, 2p$, are the $2p$ p-dimensional cubature points. The location of the points is given by:

$$\alpha_i^{2r-1} = \sqrt{\frac{2}{3}} \cos\left(\frac{(2r-1)\,i\pi}{p}\right), \quad \alpha_i^{2r} = \sqrt{\frac{2}{3}} \sin\left(\frac{(2r-1)\,i\pi}{p}\right),$$

$$r = 1, 2, \ldots, [p/2],$$

where $[p/2]$ is the greatest integer not exceeding $p/2$, and if p is odd $\alpha_i^p = (-1)^i/\sqrt{3}$.

4.4.2 Affine mappings

The Stroud cubature allows to numerically evaluate the integral of a function on the reference hypercube $[-1, 1]^p$. Usually all cubature formulas consider some reference interval of integration, so that reasonable mappings are required to make these formulas available to more general intervals. If we are dealing with interpolatory formulas, then we have to use *affine transformations*, since these transformations are the only ones that preserve the degree of a given polynomial under the mapping [6].

In fact, the general multidimensional integral

$$I = \int_{L_1}^{U_1} \int_{L_2}^{U_2} \cdots \int_{L_p}^{U_p} f(x_1, x_2, \ldots, x_p)\, \mathrm{d}x_1\, \mathrm{d}x_2 \cdots \mathrm{d}x_p$$

can be transformed into an integral over a hypercube $[-1, 1]^p$ by means of the affine transformation

$$x_i = \frac{U_i + L_i}{2} + y_i \frac{U_i - L_i}{2}, \quad i = 1, 2, \ldots, p.$$

The determinant of the Jacobian matrix of this transformation is:

$$|J| = \prod_{i=1}^{p} \left(\frac{U_i - L_i}{2} \right).$$

This transformation is very general and can be applied to any quadrature formula to generalize it to any integration intervals.

For instance, in the case of the Stroud cubature formula of degree 2, we map the $p + 1$ Stroud cubature points according to:

$$\tilde{\alpha}_i^j = \frac{U_i + L_i}{2} + \alpha_i^j \frac{U_i - L_i}{2}, \quad i = 1, 2, \ldots, p + 1 \text{ and } j = 1, 2, \ldots, p.$$

4.4.3 Univariate Gauss Formulas

In this section we review some results for univariate integration formulas and in the next we will switch to multivariate integration.

We consider the definite integral of the function $f(x)$ on the interval $[a, b]$

$$I[f] = \int_a^b f(x)\, \mathrm{d}x,$$

and define its numerical approximation as follows.

Definition 4.4. *A weighted sum of function values*

$$Q_n(f) = \sum_{i=1}^{n} w_i\, f(x_i)$$

is called a numerical n-point integration formula *if it is in some sense an approximation of* $I[f]$*. The sampling points* $x_1, \ldots, x_n$ *are called integration abscissas and the values* $w_1, \ldots, w_n$ *are called integration weights. The abscissas and weights of an integration formula must be independent of the function* f*.*

The way we choose the integration abscissas defines uniquely a certain class of interpolatory formulas. For instance, one can choose the integration abscissas as the *zeros of some classical orthogonal polynomials*. In this case, we end up with the so-called *Gauss quadrature formulas*.

The following theorem concerns the *degree of exactness* provided by the Gauss quadrature formulas.

Theorem 4.3. *The degree of exactness of an* n*-point quadrature formula*

$$Q_n f = \sum_{i=1}^{n} w_i \, f(x_i)$$

is $2n-1$*. This degree of exactness can be obtained by using the zeros of the* n*-th orthogonal polynomial of degree* n *in* $[a, b]$ *as the integration abscissas* $x_1, \ldots, x_n$ *of the interpolatory formula.*

As for the orthogonal polynomials, there are several possible choices, such as Legendre polynomials, Laguerre polynomials, Hermite polynomials or Jacobi polynomials. If we choose the Legendre polynomial of degree n, the integration abscissas $x_1, \ldots, x_n$ are the zeros of this Legendre polynomial. The quadrature formulas obtained by making this choice are called *Gauss-Legendre formulas*, and hereinafter they will be denoted by G_n.

The usual domain of definition of the orthogonal polynomials is $\bar{B} = [-1, 1]$. This means that if $[a, b]$ is an arbitrary interval we need to map the integration abscissas $\bar{x}_i$ (i.e., the zeros of the orthogonal polynomial) according to the affine transformation

$$x_i = \bar{x}_i \, \frac{b-a}{2} + \frac{a+b}{2}, \quad i = 1, 2, \ldots, n,$$

and the corresponding integration weights are mapped according to

$$w_i = \bar{w}_i \, \frac{b-a}{2}.$$

Compare with what we have seen in 4.4.2, and observe that *here, differently from the Stroud cubature, there are also integration weights to be mapped.*

4.4.4 Construction of multivariate formulas by product rules

When dealing with one-dimensional integration, we know that we can count on a wide variety of quadrature formulas, but to work out the ANOVA expansion we need to be able to perform numerical integration of multidimensional functions, as we have already pointed out. The simplest approach to extend the many known univariate integration formulas to the case of multiple dimensions is by means of

Cartesian products and product rules [6]. Let us consider $B_1 \subseteq \mathbb{R}^{d_1}$, a region in d_1-dimensional Euclidean space with points $\boldsymbol{x^1} \equiv (x_1^1, x_2^1, \ldots, x_{d_1}^1)$, and let $B_2 \subseteq \mathbb{R}^{d_2}$ be a region in d_2-dimensional Euclidean space with points $\boldsymbol{x^2} \equiv (x_1^2, x_2^2, \ldots, x_{d_2}^2)$. The notation $B_1 \times B_2$ indicates the *Cartesian product of* B_1 *and* B_2, namely the region in the Euclidean space of $d_1 + d_2$ dimensions with points $(\boldsymbol{x^1}, \boldsymbol{x^2})$ that satisfy $\boldsymbol{x^1} \in B_1$ and $\boldsymbol{x^2} \in B_2$.

Now if $Q_{n_1}^1$ is an n_1-point univariate integration formula over B_1,

$$Q_{n_1}^1(f_1) = \sum_{i_1=1}^{n_1} w_{i_1}^1 f_1(\boldsymbol{x}_{i_1}^{\boldsymbol{1}}) \approx \int_{B_1} f_1(\boldsymbol{x^1})\,\mathrm{d}\boldsymbol{x^1}, \quad \boldsymbol{x}_{i_1}^{\boldsymbol{1}} \in B_1,$$

and if $Q_{n_2}^2$ is an n_2-point univariate integration formula over B_2,

$$Q_{n_2}^2(f_2) = \sum_{i_2=1}^{n_2} w_{i_2}^2 f_2(\boldsymbol{x}_{i_2}^{\boldsymbol{2}}) \approx \int_{B_2} f_2(\boldsymbol{x^2})\,\mathrm{d}\boldsymbol{x^2}, \quad \boldsymbol{x}_{i_2}^{\boldsymbol{2}} \in B_2,$$

then with the name *product rule of* $Q_{n_1}^1$ *and* $Q_{n_2}^2$ we designate the $n_1 n_2$-point rule applicable to $B_1 \times B_2$ and defined by

$$(Q_{n_1}^1 \times Q_{n_2}^2)(f) := \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} w_{i_1}^1 w_{i_2}^2 f(\boldsymbol{x}_{i_1}^{\boldsymbol{1}}, \boldsymbol{x}_{i_2}^{\boldsymbol{2}}) \approx \int_{B_1 \times B_2} f(\boldsymbol{x^1}, \boldsymbol{x^2})\,\mathrm{d}\boldsymbol{x^1}\mathrm{d}\boldsymbol{x^2}.$$

We point out that $\boldsymbol{x}_{i_1}^{\boldsymbol{1}}$ and $\boldsymbol{x}_{i_2}^{\boldsymbol{2}}$ are the *integration abscissas* for $\boldsymbol{x^1}$ and $\boldsymbol{x^2}$, respectively. The following theorem is of interest to us [6, 19].

Theorem 4.4. *If* $Q_{n_1}^1$ *integrates* $f_1(\boldsymbol{x^1})$ *exactly over* B_1*, and if* $Q_{n_2}^2$ *integrates* $f_2(\boldsymbol{x^2})$ *exactly over* B_2*, then the product rule* $Q_{n_1}^1 \times Q_{n_2}^2$ *will integrate their product*

$$f(\boldsymbol{x^1}, \boldsymbol{x^2}) := f_1(\boldsymbol{x^1})\, f_2(\boldsymbol{x^2}), \quad \boldsymbol{x^1} \in B_1,\ \boldsymbol{x^2} \in B_2,$$

exactly over the region $B := B_1 \times B_2$.

The product rule can be generalized to higher dimensions in a straightforward fashion. Assume that the integration region B is a Cartesian product of $p \geq 3$ regions $B_1, \ldots, B_p$, i.e. $B = B_1 \times \cdots \times B_p$. Let $Q_{n_k}^k$ denote the n_k-point univariate integration formulas over B_k

$$Q_{n_k}^k(f_k) = \sum_{i_k=1}^{n_k} w_{i_k}^k f_k(\boldsymbol{x}_{i_k}^{\boldsymbol{k}}) \approx \int_{B_k} f_k(\boldsymbol{x^k})\,\mathrm{d}\boldsymbol{x^k}, \quad k = 1, \ldots, p,$$

then the *product rule* $(Q_{n_1}^1 \times \cdots \times Q_{n_p}^p)(f)$ is defined by

$$\begin{aligned}(Q_{n_1}^1 \times \cdots \times Q_{n_p}^p)(f) &:= \sum_{i_1=1}^{n_1} \cdots \sum_{i_p=1}^{n_p} w_{i_1}^1 \cdots w_{i_p}^p f(\boldsymbol{x}_{i_1}^{\boldsymbol{1}}, \ldots, \boldsymbol{x}_{i_p}^{\boldsymbol{p}}) \\ &\approx \int_B f(\boldsymbol{x^1}, \ldots, \boldsymbol{x^p})\,\mathrm{d}\boldsymbol{x^1} \cdots \mathrm{d}\boldsymbol{x^p}.\end{aligned}$$

Usually product rules $Q^1_{n_1} \times \cdots \times Q^p_{n_p}$ whose constituent univariate integration formulas are identical,

$$Q^1_{n_1} = Q^2_{n_2} = \cdots = Q^p_{n_p} = Q_n,$$

are denoted as $(Q_n)^p$. For instance, let G_5 be the univariate Gauss-Legendre integration rule over $[-1, 1]$ (Section 4.4.3). Then $G_5 \times G_5 \times G_5 \times G_5$ $(= (G_5)^4)$ is a product rule of 625 points applicable to the 4-dimensional hypercube $[-1, 1]^4$. It integrates exactly the 10^4 monomials of the form $x_1^{n_1} x_2^{n_2} x_3^{n_3} x_4^{n_4}$ with $0 \leq n_1, n_2, n_3, n_4 \leq 9$.

We must emphasize that here we are entering in dangerous waters, since the product formula $(Q_n)^p$ has n^p integration abscissas. It is apparent that if the dimension of the function to be integrated is increased by one, then the number of integration abscissas is increased by a factor of n. This rapidly increasing computational cost makes the use of product formulas virtually impracticable unless we are dealing with a moderate dimensionality (typically $p \leq 5$).

In the tests that follow and in the application of the ANOVA expansion to the BGK+PML model we will deal with functions having at most 6 dimensions. With moderate numbers of dimensions it is still practicable to use the product rules with Gauss-Legendre quadrature formulas. If instead one is willing to increase the efficiency of the algorithm while at the same keeping the accuracy of the numerical integration, then the sparse grid integration methods are really the way to go; see [10] for the details.

4.5 Some preliminary tests

Before applying to our BGK+PML model all the machinery that we have discussed so far, we want to evaluate the calculation of the ANOVA expansion for some simpler benchmark functions. In this section we provide some results for the ANOVA expansion and the calculation of the TSIs for a subset of the classic test functions [12, 13]. We consider the following test functions:

- Product Peak function: $f_1(\boldsymbol{x}) = \prod_{i=1}^{p} \left[c_i^{-2} + (x_i - \omega_i)^2\right]^{-1}$,
- Gaussian function: $f_2(\boldsymbol{x}) = \exp\left[-\sum_{i=1}^{p} c_i^2 (x_i - \omega_i)^2\right]$,

where the coefficients $\boldsymbol{c} = (c_1, \ldots, c_p)$ and $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_p)$ are generated randomly.

The plots in Figure 4.6 show the L^2-norm of the error between the 4-dimensional Gaussian function and its truncated ANOVA expansions as a function of the truncation order, where the numerical integration has been carried out according to several quadrature formulas that we have discussed in the previous section. In Figures 4.6a and 4.6b the Stroud cubatures of degree 2 and 3 have been used, respectively. To obtain Figures 4.6c and 4.6d we built multivariate integration formulas by applying the product rule to the Gauss-Legendre quadrature formula with 2 and 3 integration abscissas for each dimension, respectively. In all the cases, the variables $\boldsymbol{x}$ are assumed to vary in the interval $[-1, 1]$.

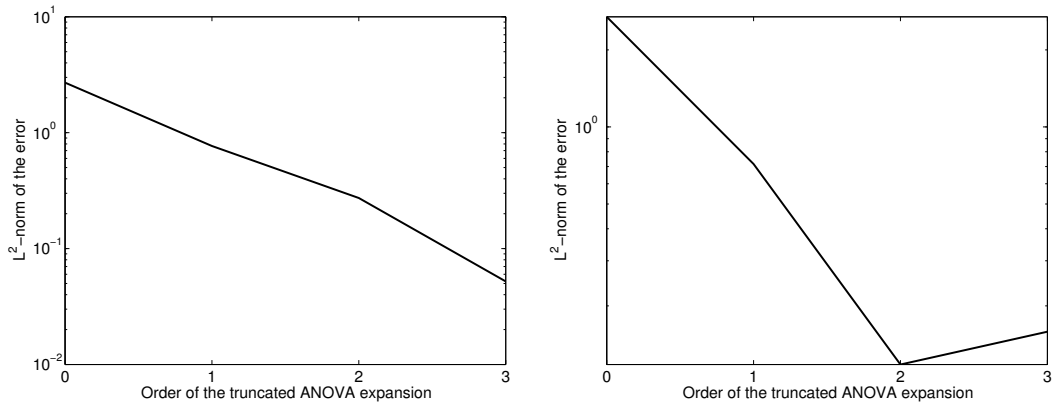
The four integration techniques above have also been used to compute the TSIs for the variables of the 4-dimensional Gaussian that we considered in Section 4.3.3. The TSI values are reported in Table 4.1, together with the exact values computed with *Mathematica*®.

	x_1	x_2	x_3	x_4
Stroud-2	0.0263	0.0497	0.9291	0.0523
Stroud-3	0.0096	0.0284	0.9400	0.0583
$(G_2)^4$	0.0008	0.0200	0.9604	0.0405
$(G_3)^4$	0.0010	0.0249	0.9453	0.0600
Exact	0.0009	0.0239	0.9428	0.0598

Table 4.1: TSIs for a 4D Gaussian test function, computed according to different integration formulas, and exact values.

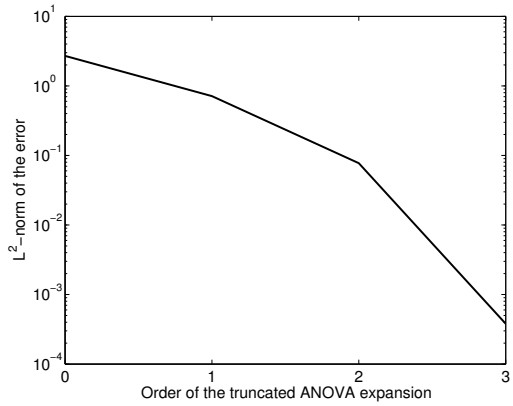
It is clear from the table that $(G_3)^4$ is extremely good at approximating the TSIs, but it also appears that Stroud-3 gives already a good indication about the importance of the single variables.

However, if we look at Figure 4.6b it appears that Stroud-3 does not yet give a nice convergence behaviour. What we are really looking for here is a trade-off between reasonable efficiency and reasonable accuracy.

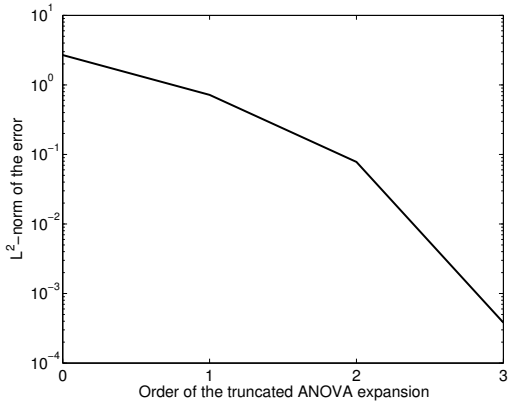


(a) Stroud-2

(b) Stroud-3



(c) $(G_2)^4$



(d) $(G_3)^4$

Figure 4.6: Convergence behaviour of the truncated ANOVA expansion of a 4D Gaussian test function for four different numerical integration formulas.

In the following pages we go to higher dimensions and consider the 6D Product Peak function and the 6D Gaussian function having the coefficients

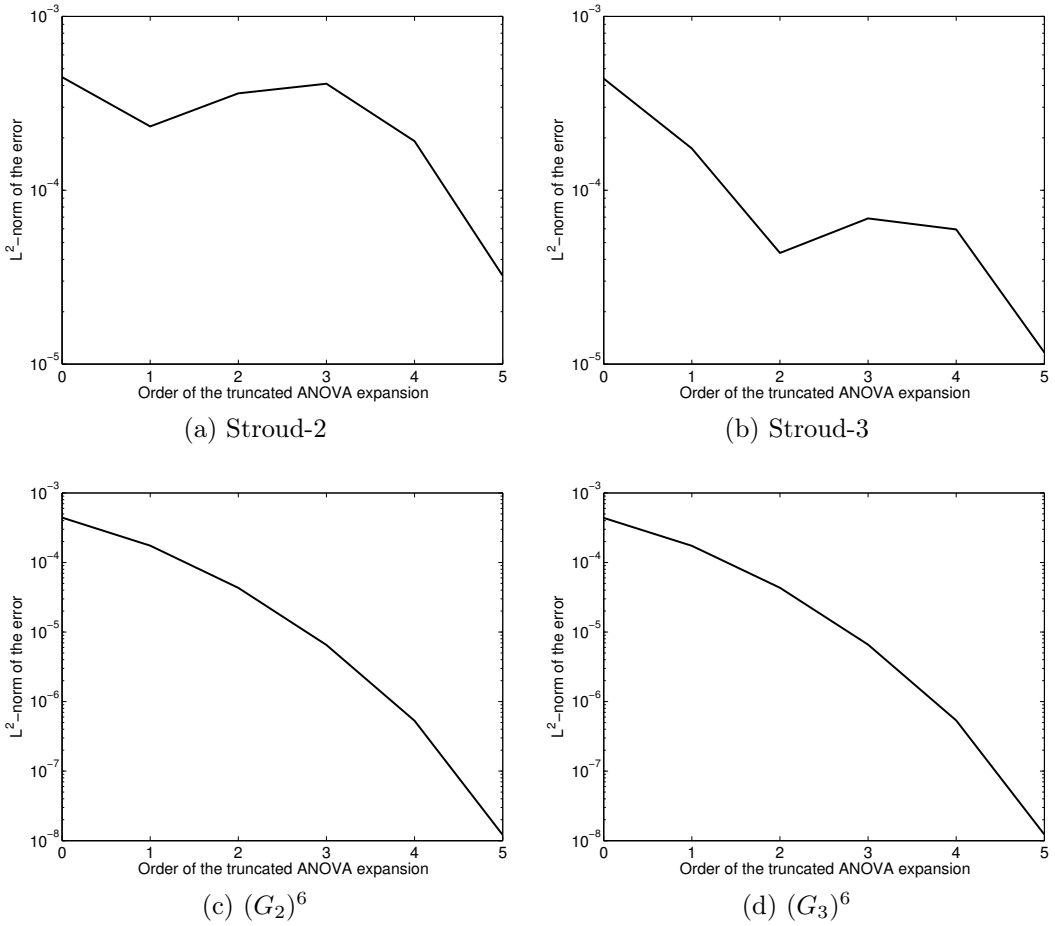
$$\boldsymbol{c} = (0.122, 0.3627, 0.908, 0.5756, 0.5349, 0.7462),$$

$$\boldsymbol{\omega} = (0.989, 0.567, 0.9898, 0.3223, 0.8135, 0.5157).$$

The convergence behaviour of the ANOVA expansion according to its truncation order is illustrated in Figures 4.7 and 4.8 for the Product Peak function and the Gaussian function, respectively. Again, from Figures 4.7a, 4.7b, 4.8a and 4.8b it is evident that Stroud-2 and Stroud-3 are not enough to obtain a correct convergence behaviour. On the contrary, we can observe from Figures 4.7c, 4.7d, 4.8c and 4.8d that the contributions to the error become monotonically smaller and smaller as the truncation order of the ANOVA expansion increases. Moreover, it is encouraging that the convergence behaviour appears to be insensitive to the choice of test function.

Tables 4.2 and 4.3 confirm that Stroud-2 is not accurate at all, while Stroud-3 appears to be misleading with regard to the TSIs of x_1 and x_2, even though yielding reasonable results for the other TSI values. We must use Gauss-Legendre quadrature if we want to be sure to obtain reliable results for the TSIs.

On the basis of such considerations, in what follows we will only make use of the product rules with Gauss-Legendre integration formulas.



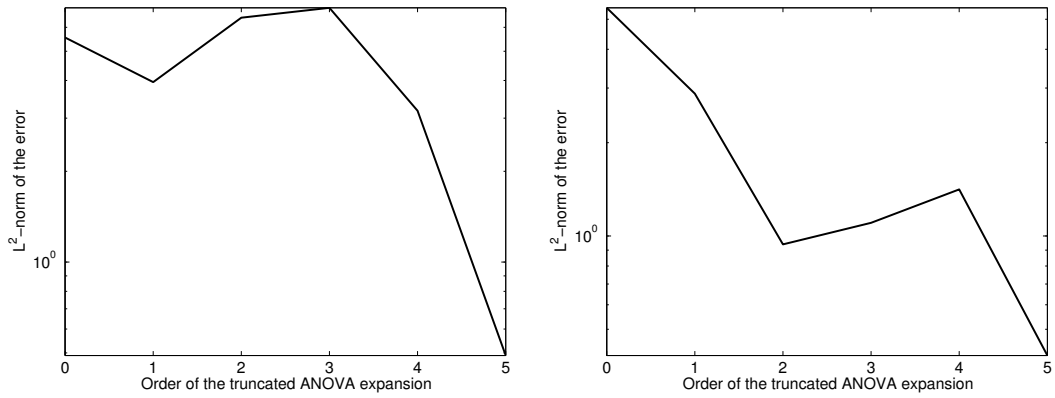
(a) Stroud-2 (b) Stroud-3

(c) $(G_2)^6$ (d) $(G_3)^6$

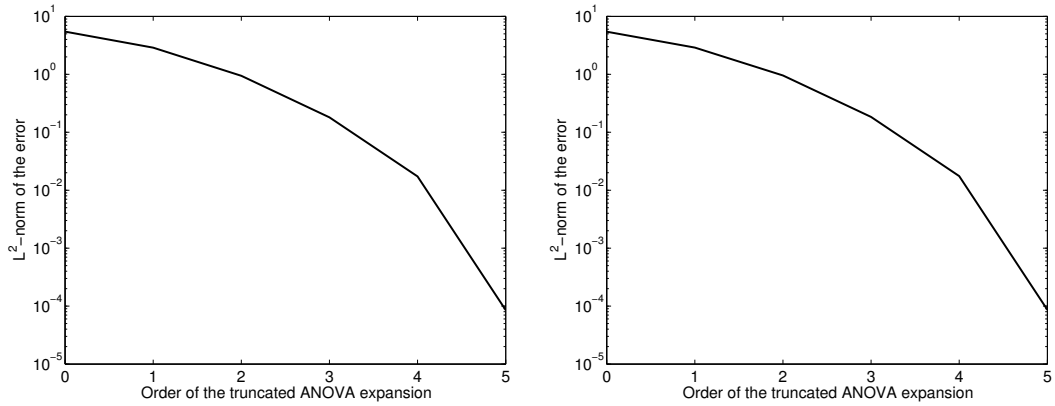
Figure 4.7: Convergence behaviour of the truncated ANOVA expansion of a 6D Product Peak test function for four different numerical integration formulas.

	x_1	x_2	x_3	x_4	x_5	x_6
Stroud-2	0.3135	0.3143	0.5925	0.2266	0.2229	0.2267
Stroud-3	0.0243	0.0500	0.6182	0.0765	0.1720	0.2187
$(G_2)^6$	0.0011	0.0236	0.6426	0.0434	0.1586	0.2201
$(G_3)^6$	0.0012	0.0270	0.6582	0.0573	0.1546	0.1958
$(G_4)^6$	0.0013	0.0281	0.6495	0.0599	0.1573	0.2031

Table 4.2: TSIs for a 6D Product Peak function, computed according to different integration formulas.



(a) Stroud-2 (b) Stroud-3



(c) $(G_2)^6$ (d) $(G_3)^6$

Figure 4.8: Convergence behaviour of the truncated ANOVA expansion of a 6D Gaussian test function for four different numerical integration formulas.

	x_1	x_2	x_3	x_4	x_5	x_6
Stroud-2	0.3770	0.3567	0.6322	0.2286	0.2497	0.2517
Stroud-3	0.0527	0.0734	0.7112	0.0932	0.1783	0.2166
$(G_2)^6$	0.0006	0.0159	0.7610	0.0321	0.1395	0.2010
$(G_3)^6$	0.0008	0.0197	0.7494	0.0476	0.1461	0.2013
$(G_4)^6$	0.0008	0.0208	0.7450	0.0505	0.1499	0.2064

Table 4.3: TSIs for a 6D Gaussian test function, computed according to different integration formulas.

4.6 Flowcharts of the ANOVA expansion code

Figure 4.9 shows a flowchart for the `ComputeTSI` function. According to the user's choice, either the function `GenerateStroudPoints` or `GenerateGaussLegendreNodes` is used to generate the cubature nodes needed to calculate the TSIs. Then inside the loop the `ComputeANOVA` function is called for each cubature node to calculate the $V_{\mathcal{T}}$ values. After this, the sensitivity measures $S_{\mathcal{T}}$ and finally the TSIs are computed.

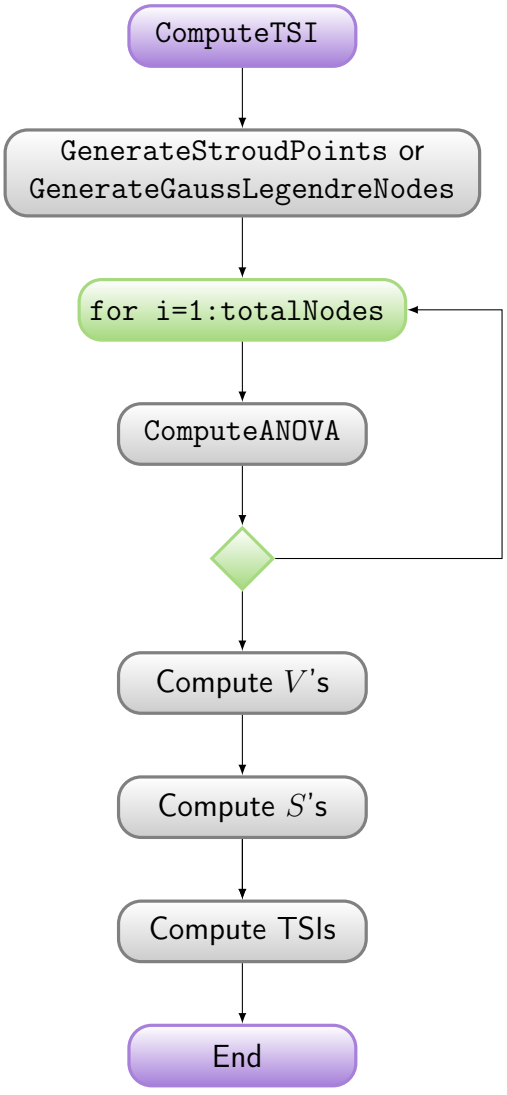


Figure 4.9: Flowchart of the ComputeTSI function.

Figure 4.10 illustrates the content of the `ComputeANOVA` function. First, the zero-th order term u_0 of the expansion is computed using the `Compute_u0` function. Then the outer loop cycles over the dimensions of the problem, from $t = 1$ to $t = p - 1$. The inner loop cycles over the terms of order t, which are $\binom{p}{t}$-many, and at each iteration calls the function `ComputeIntegral`. Finally we subtract from the higher order terms of the ANOVA expansion the contributions coming from the lower order terms.

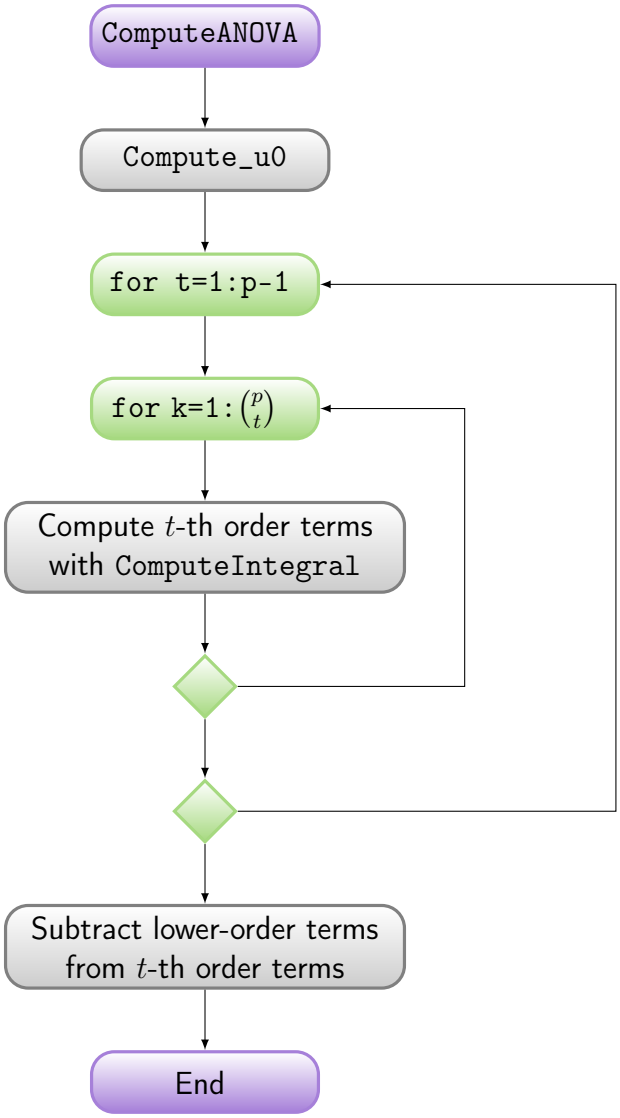


Figure 4.10: Flowchart of the ComputeANOVA function.

Figure 4.11 reports the flowchart of the `Compute_u0` function. To generate the cubature nodes needed to calculate u_0 one can choose either the `GenerateStroudPoints` or the `GenerateGaussLegendreNodes` function. Inside the loop, for each cubature node we call `TestFunction` if we are using a test function or `ErrorFunctional` if we are applying the ANOVA expansion machinery to the BGK+PML model. The calculation of the integral is finalized by using an affine mapping and the zero-th order term of the ANOVA expansion is returned.

We point out that the `ComputeIntegral` function develops similarly to the `Compute_u0` function. Additional documentation is contained in the code.

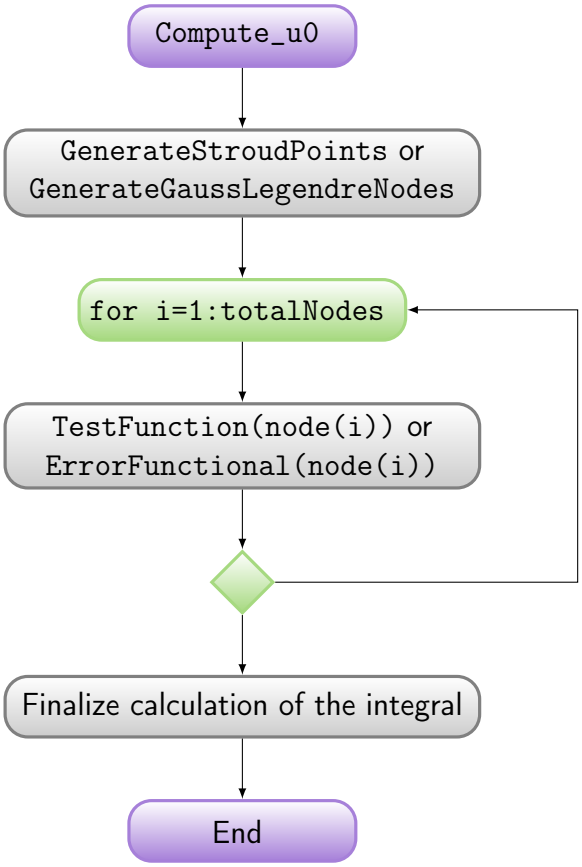


Figure 4.11: Flowchart of the Compute_u0 function.

4.7 ANOVA expansion applied to the BGK+PML model

Finally we have reached the heart of this study, since we now have all the ingredients to apply the ANOVA expansion to the functional of our solution to the BGK+PML model. We have already delineated our functional at the beginning of this chapter, but let us recall it again here. We choose the functional g_1 to be the maximum over the simulation time of the L^2-norm of the error in the density a_0 between the BGK+PML and the plain BGK, calculated along a line close to the PML, normalized with respect to the L^2-norm of the initial condition of the density a_0 on that same line for the plain BGK. Synthetically:

$$g_1 = \frac{\max_{t\in[0,T]} \left\{ \left[\int_0^{L_y} \left(a_0^{\mathrm{BGK+PML}}(x^*, y, t) - a_0^{\mathrm{BGK}}(x^*, y, t) \right)^2 \mathrm{d}y \right]^{1/2} \right\}}{\left[\int_0^{L_y} \left(a_0^{\mathrm{BGK}}(x^*, y, t = 0) \right)^2 \mathrm{d}y \right]^{1/2}},$$

We consider g_1 to be a function of α_0, α_1 and L, namely $g_1(\alpha_0, \alpha_1, L)$. We recall that we must freeze λ_0 and λ_1 to zero to address the stability issues discussed in Sections 3.3.1 and 4.2.1.

The TSIs are computed for $\beta = 2, 3, 4$, being β the exponent that appears in the expression of the damping function σ_1. We allow both α_0 and α_1 to vary in the interval $[0, 5]$, while the PML thickness L varies in the interval $[0.25, 0.80]$. The results, obtained using increasingly more accurate cubature formulas, namely $(G_2)^3$, $(G_3)^3$ and $(G_4)^3$, are given in Table 4.4 for $\beta = 2, 3, 4$.

PML exponent	Cubature type	α_0	α_1	L
$\beta = 2$	$(G_2)^3$	0.2274	0.2521	0.9435
	$(G_3)^3$	0.2251	0.2565	0.9478
	$(G_4)^3$	0.2221	0.2494	0.9607
$\beta = 3$	$(G_2)^3$	0.2212	0.2511	0.9586
	$(G_3)^3$	0.2104	0.2590	0.9716
	$(G_4)^3$	0.2112	0.2491	0.9705
$\beta = 4$	$(G_2)^3$	0.2201	0.2352	0.9640
	$(G_3)^3$	0.2114	0.2488	0.9814
	$(G_4)^3$	0.2051	0.2427	0.9865

Table 4.4: TSIs for the parameters α_0, α_1 and L, using functional $g_1(\alpha_0, \alpha_1, L)$.

It is evident from Table 4.4 that the parameter with the largest TSI is the PML thickness L, while α_0 and α_1 basically have the same sensitivity measure. Moreover, we observe that the TSIs are virtually independent of the PML exponent β.

We can also take into account the PML exponent as a parameter of the functional, i.e. we consider $g_1(\alpha_0, \alpha_1, \beta, L)$. We compute the TSIs assuming the following intervals of variation for the parameters

$$\alpha_0 \in [0, 3.5]\,, \quad \alpha_1 \in [0, 3.5]\,, \quad \beta \in [0, 4]\,, \quad L \in [0.25, 0.80]\,.$$

The results are reported in Table 4.5, where we have highlighted the values obtained with the most accurate cubature formula we used.

Cubature type	α_0	α_1	β	L
$(G_2)^4$	0.1638	0.2474	0.2775	0.9312
$(G_3)^4$	0.1635	0.1916	0.2879	0.9385

Table 4.5: TSIs for the parameters α_0, α_1, β and L, using functional $g_1(\alpha_0, \alpha_1, \beta, L)$.

We observe, as we might have expected, that the parameters α_0 and α_1 show only a minor impact on the functional g_1. The parameters β and L appearing in the definition of the damping function σ_1 clearly dominate the outcome of the functional. In particular, the PML thickness L has the largest influence on g_1.

At this point, it might be interesting to investigate what happens if we freeze the least important parameters $\alpha_0 = \alpha_1 = 1$ and we do the ANOVA expansion of the functional $g_1(\beta, L)$. The TSI values for this case are reported in Table 4.6, where again we have highlighted the most accurate calculations.

Cubature type	β	L
$(G_2)^2$	0.0465	0.9535
$(G_3)^2$	0.0557	0.9443
$(G_4)^2$	0.0660	0.9340

Table 4.6: TSIs for the parameters β and L, using functional $g_1(\beta, L)$.

The TSI values in Table 4.6 strongly confirm that the PML thickness L is the parameter having the largest influence on the PML behaviour.

4.7.1 Other functionals

In this section we also consider other forms of the error functional to investigate whether this may have an influence on the TSI values.

We define the error functional g_2 as

$$g_2 = \frac{\int_0^T \left\{ \left[\int_0^{L_y} \left(a_0^{\mathrm{BGK+PML}}(x^*, y, t) - a_0^{\mathrm{BGK}}(x^*, y, t) \right)^2 \, \mathrm{d}y \right]^{1/2} \right\} \mathrm{d}t}{\left[\int_0^{L_y} \left(a_0^{\mathrm{BGK}}(x^*, y, t = 0) \right)^2 \, \mathrm{d}y \right]^{1/2}},$$

Here we are still looking at the error in the density a_0 on a vertical line close to the PML, but instead of taking the maximum of the L^2-norm we compute its integral over time. The functional g_2 is basically the area under the curve in Figure 4.2.

Table 4.7 reports the values obtained for the TSIs when adopting the functional $g_2(\alpha_0, \alpha_1, \beta, L)$.

Cubature type	α_0	α_1	β	L
$(G_2)^4$	0.1623	0.1875	0.3061	0.9343
$(G_3)^4$	0.1596	0.1659	0.3740	0.9298

Table 4.7: TSIs for the parameters α_0, α_1, β and L, using functional $g_2(\alpha_0, \alpha_1, \beta, L)$.

It is encouraging to observe that these values are very similar to those in Table 4.5, meaning that the sensitivity of the model to the various parameters is virtually independent of the choice of the functional. The only noticeable effect produced by switching from functional g_1 to functional g_2 is that the TSI of α_1 decreases a bit, while the TSI of β increases.

Freezing $\alpha_0 = \alpha_1 = 1$ and redoing the ANOVA analysis using the functional $g_2(\beta, L)$, we obtain the results in Table 4.8.

Cubature type	β	L
$(G_2)^2$	0.0543	0.9457
$(G_3)^2$	0.0803	0.9197
$(G_4)^2$	0.0924	0.9076

Table 4.8: TSIs for the parameters β and L, using functional $g_2(\beta, L)$.

Also here we can note the similarities with the TSI values in Table 4.6, and conclude that the TSIs do not significantly depend on the choice of the functional.

Another option for the error functional is to calculate the L^2-norm of the error in a_0 not only on a vertical line close to the PML, but on the entire domain $L_x \times L_y$, and then integrate over time:

$$g_3 = \frac{\int_0^T \left\{ \left[\int_0^{L_x} \int_0^{L_y} \left(a_0^{\mathrm{BGK+PML}}(x,y,t) - a_0^{\mathrm{BGK}}(x,y,t) \right)^2 \, \mathrm{d}x \, \mathrm{d}y \right]^{1/2} \right\} \mathrm{d}t}{\left[\int_0^{L_x} \int_0^{L_y} \left(a_0^{\mathrm{BGK}}(x,y,t=0) \right)^2 \, \mathrm{d}x \, \mathrm{d}y \right]^{1/2}},$$

This choice of the functional yields the TSI values reported in the following table.

Cubature type	α_0	α_1	β	L
$(G_2)^4$	0.1649	0.1773	0.3072	0.9374
$(G_3)^4$	0.1605	0.1627	0.3920	0.9280

Table 4.9: TSIs for the parameters α_0, α_1, β and L, using functional $g_3(\alpha_0, \alpha_1, \beta, L)$.

The results in Table 4.9 again confirm that the TSIs are basically independent of the choice of the error functional.

The results in Table 4.10, obtained for the functional $g_3(\beta, L)$, are also in agreement with our previous observations.

Cubature type	β	L
$(G_2)^2$	0.0533	0.9467
$(G_3)^2$	0.0845	0.9155
$(G_4)^2$	0.1053	0.8947

Table 4.10: TSIs for the parameters β and L, using functional $g_3(\beta, L)$.

Summarizing, we have answered the question *"What are the most important parameters in the BGK+PML model?"*. For stability issues, we know that we have to freeze λ_0 and λ_1, while the results of the sensitivity analysis show that the most significant parameters are the PML exponent β and the PML thickness L.

In the light of these results, it is now tempting to ask the question *"How can the most significant parameters be chosen in an optimal way?"*, where *in an optimal way* can be restated as *so as to minimize the functional* g.

4.8 Choice of the optimal parameter values

In order to choose the optimal parameter values, we resort to convex optimization techniques. In general, optimization techniques are employed to find a set of design variables, $\boldsymbol{\alpha} = (\alpha^1, \alpha^2, \cdots, \alpha^p) \in \mathbb{R}^p$, that can in some way be defined as optimal. In a simple case this might be the minimization or maximization of some objective function $g(\boldsymbol{\alpha})$. In a more advanced formulation the objective function might be subject to constraints in the form of equality constraints, inequality constraints, and/or parameter bounds.

We can state the general minimization problem as

$$\min_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha}).$$

If we want to find an efficient and accurate solution to this problem, we have to take into account the nature of the objective function and the constraints. If the objective function and the constraints are both linear functions, the problem is known as a linear programming problem. Quadratic programming deals with the minimization or maximization of a quadratic objective function that is linearly constrained. For both linear and quadratic programming problems, reliable solution procedures are readily available. If the objective function and/or the constraints are nonlinear functions of the design variables, then one has to solve a nonlinear programming problem. A nonlinear programming problem is in general more difficult to solve and needs an iterative procedure to choose a direction of search at each iteration. This is usually achieved by the solution of a linear programming, a quadratic programming or an unconstrained subproblem.

In our case we could introduce as a constraint a function representing the computational cost. Yet we already know that the computational cost depends on the mesh-size Δx and on the PML thickness L. Since we are not changing the mesh-size Δx, we know that if L increases then the computational cost will also increase. So we can perform an unconstrained minimization of $g_1(\alpha_0, \alpha_1, \beta, L)$ on the domain $(\alpha_0, \alpha_1, \beta, L) \in [0, 3.5] \times [0, 3.5] \times [0, 4] \times [0.25, 0.80]$.

Table 4.11 reports four set of parameter values that minimize the functional g_1. These optimal parameter values have been found through a minimization procedure starting from four different initial guesses.

α_0	α_1	β	L
2.7561	2.7361	3.3077	0.6717
2.5493	2.0772	3.8463	0.5505
0.4991	0.4749	3.8877	0.4222
0.2551	0.0609	3.9325	0.4133

Table 4.11: Four sets of optimal values for the parameters α_0, α_1, β and L, obtained by minimizing the functional $g_1(\alpha_0, \alpha_1, \beta, L)$.

It appears from Table 4.11 that there are many combinations of parameter values that minimize the functional g_1. The parameters α_0 and α_1 can basically take any value since they show no significant impact on the outcome of the minimization of the functional g_1. This is in agreement with the fact that their TSI values are the lowest ones, as we have seen in Section 4.7. The other two parameters, β and L, seem to take more definite values. In particular, it seems that when β increases, L decreases, and vice-versa.

To further investigate this aspect, we plot the surface of $g_1(\beta, L)$ on the domain $(\beta, L) \in [0, 4] \times [0.10, 1.00]$.

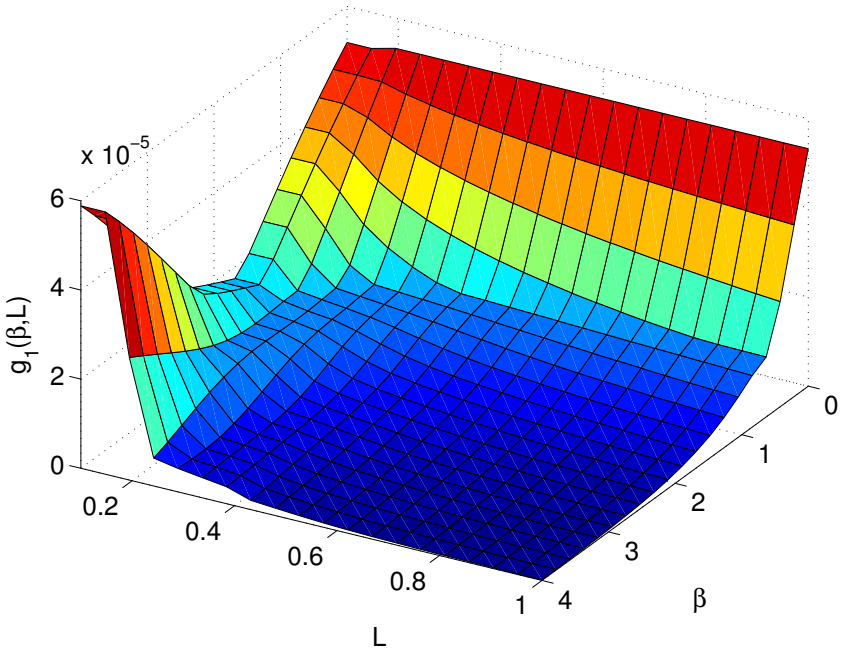


Figure 4.12: Plot of $g_1(\beta, L)$ for $(\beta, L) \in [0, 4] \times [0.10, 1.00]$.

We can observe from Figure 4.12 that the error g_1 starts to be acceptable when $\beta \gtrapprox 1$ and $L \gtrapprox 0.20$. It seems that there is a region, roughly for $L \gtrapprox 0.40$ and $\beta \gtrapprox 1.5$, in which the error g_1 is in the order of 10^{-7}.

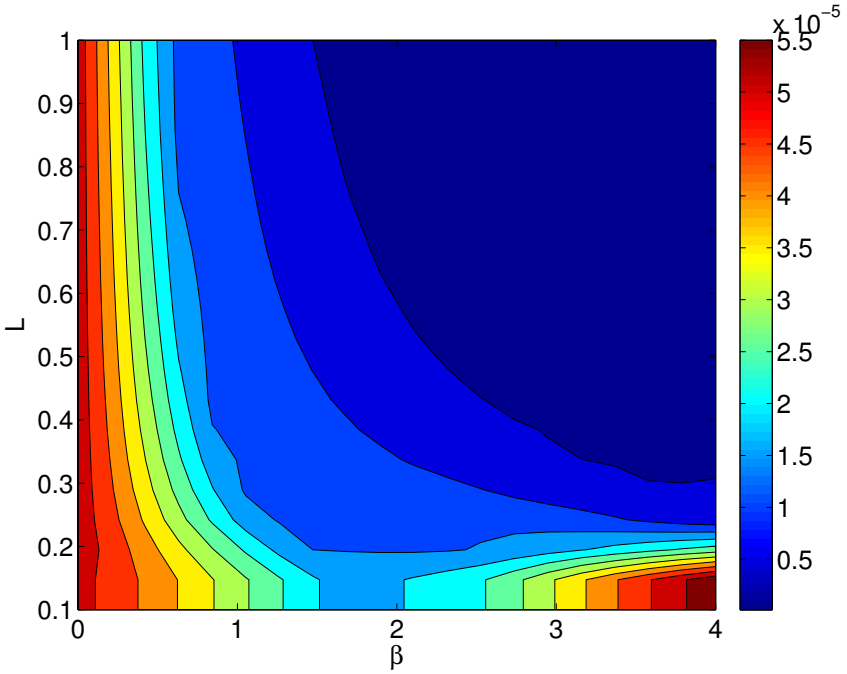
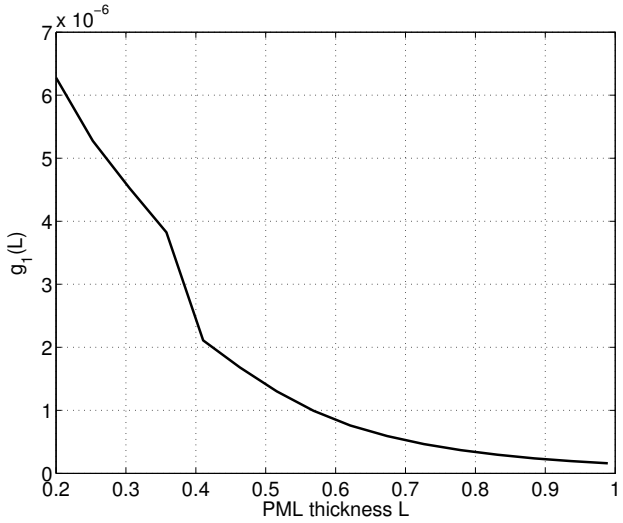


Figure 4.13: Contour plot of $g_1(\beta, L)$ for $(\beta, L) \in [0, 4] \times [0.10, 1.00]$.

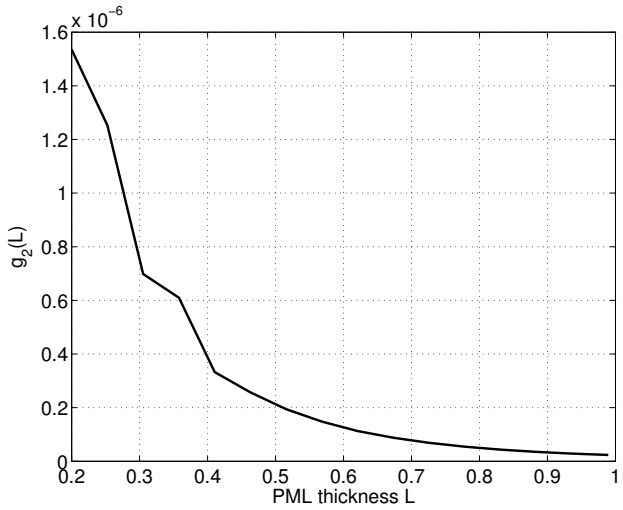
To make things clearer we report in Figure 4.13 a contour plot of $g_1(\beta, L)$ for $(\beta, L) \in [0, 4] \times [0.10, 1.00]$. It is evident from this plot that there is a region (corresponding to the dark-blue color) in which the error g_1 is practically zero. If we look at the boundary of this region, we note that when β increases, L can decrease and still g_1 remains in the order of 10^{-7}. This is in agreement with the observations we did about the results in Table 4.11.

4.8.1 g as function of L only

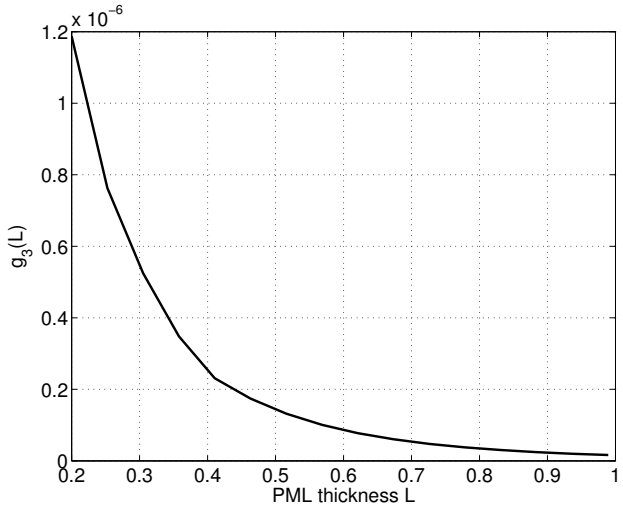
We have seen in Section 4.7 that the PML thickness L is the parameter having the highest sensitivity measure. It is tempting to discard all the other parameters and look for a simple relationship between the error and the PML thickness that can provide us with additional insights into the PML behaviour. We consider all the error functionals previously defined, namely g_1, g_2 and g_3. We run simulations with the PML thickness L varying from 0.20 to 1.00, keeping β fixed to 4, and we monitor the error functionals. The resulting plots for $g_1(L)$, $g_2(L)$ and $g_3(L)$ are shown in Figures 4.14a, 4.14b and 4.14c, respectively. All the plots show that as L increases, one gets better and better accuracy. In particular, at $L \approx 0.42$ there has already been a nice drop in the error functionals g_1 and g_2 (Figures 4.14a and 4.14b, respectively). The functional g_3, instead, shows a smoother behaviour. If one goes further than $L \approx 0.42$, then of course the accuracy improves but the computational cost becomes dominant. After some point, say $L \approx 0.80$, it makes no sense, both from an accuracy and a computational cost point of view, to further increase L, because the very small improvements in accuracy do not justify the higher computational cost.



(a) $g_1(L)$



(b) $g_2(L)$



(c) $g_3(L)$

Figure 4.14: The error functionals $g_1(L)$, $g_2(L)$ and $g_3(L)$ versus the PML thickness L.

4.8.2 Influence of the initial condition

The fact that the error functional $g_1(L)$ gets very small after $L \approx 0.42$ is probably due to the particular problem we are solving. So far we never changed the initial conditions of simulation setting, but we can expect the optimal PML thickness L to be related to the average wavelength of the peak in the initial density distribution. The idea here is to modify the initial conditions, and in particular vary the shape of that peak, to gain insights into its possible influence on the PML behaviour.

We recall the expression for the initial density distribution from Section 2.7

$$a_0(x, y, t = 0) = 2\,(p_{\text{in}} - p_{\text{out}}) \exp\left[-\varepsilon\sqrt{(x - x_0)^2 + (y - y_0)^2}\right] + 1.00,$$

where

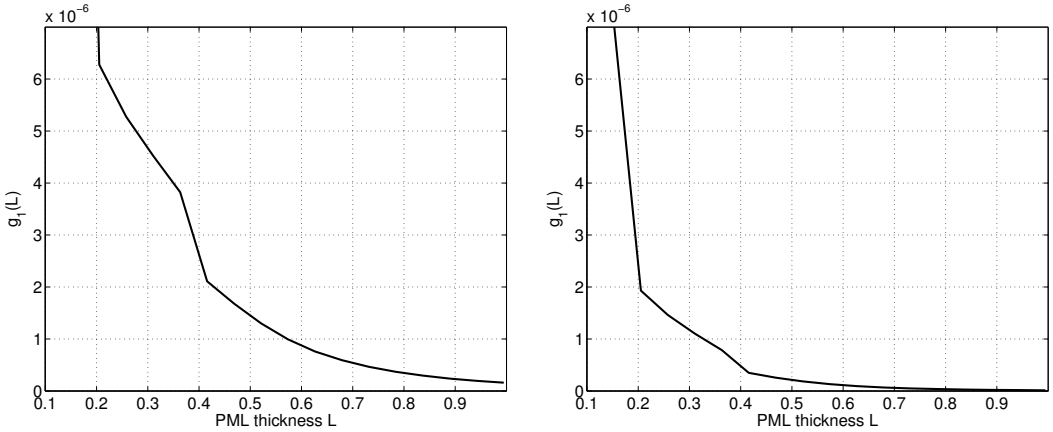
$$x_0 = L_x/2, \quad y_0 = L_y/2.$$

We can think of ε as the sharpness of the peak in the initial density distribution. In all the simulations that we have carried out so far we always used $\varepsilon = 10$. Now we assume also other values for ε, in particular we consider the set of values $\varepsilon \in \{10, 20, 30, 40\}$ and observe how ε may affect the functional $g_1(L)$.

The simulations conducted with this set of ε values were used to produce the plots reported in Figure 4.15. It is evident from Figure 4.15 that the sharpness of the peak in the initial density distribution impacts the shape of the error functional. From Figure 4.15a we can observe that at $L \approx 0.42$ the error is equal to $g_1 = 2\times10^{-6}$. If we turn our attention to the other cases, we note that this same value of g_1 is achieved

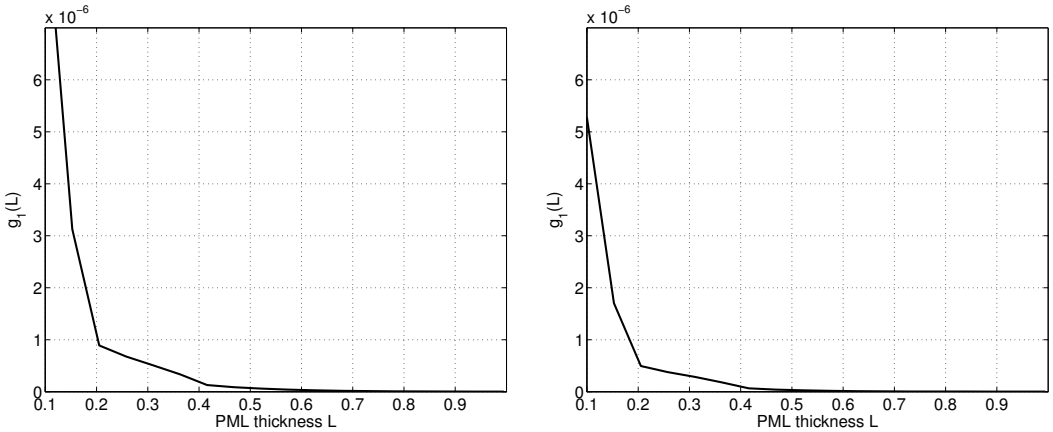
- when $L \approx 0.20$ in the case $\varepsilon = 20$;
- when $L \approx 0.18$ in the case $\varepsilon = 30$;
- when $L \approx 0.15$ in the case $\varepsilon = 40$.

We can conclude that when the peak in the initial density distribution is sharper, the PML thickness L required to attain the same level of accuracy is smaller.



(a) $\varepsilon = 10$

(b) $\varepsilon = 20$



(c) $\varepsilon = 30$

(d) $\varepsilon = 40$

Figure 4.15: Behaviour of the error functional $g_1(L)$ according to different initial conditions.

Conclusions

In this work we studied stability and optimization of an absorbing PML layer for the BGK approximation to the Boltzmann equation. We showed that for low Mach numbers and weakly compressible flows one can recover the Navier-Stokes equations from the BGK model.

We implemented a numerical scheme using fourth order accurate finite differences for the spatial discretization and a fourth order Runge-Kutta method in time. We tested the numerical scheme for a simple Couette-Poiseuille flow, for which an exact solution to the Navier-Stokes equations is available, and we verified that the BGK model is maintaining that solution.

We reviewed the theory behind the PML technique and presented a PML for the BGK model according to previous authors. We implemented this model and we showed that it is qualitatively capable of reproducing the results obtained with the plain BGK.

We investigated the role and the importance of the parameters appearing in the BGK+PML model by carrying out a stability analysis to establish reasonable bounds on the parameters. We studied the stability of this model by means of two analytical tools, both involving the symbol of a differential operator. We used the symbol to investigate the parameter values that guarantee the energy decay through time. To study the sign of the eigenvalues of the symbol we exploited a technique based on continued fraction expansion of the characteristic polynomial. We found that to ensure stability one has to freeze λ_0 and λ_1 to zero. This was also confirmed by extensive numerical simulations. The above analyses also yielded some bounds on the other parameters appearing in the BGK+PML model.

We introduced the theory of the ANOVA expansion of multivariate functions as a tool to study differential problems with a high-dimensional parameter space. We implemented the ANOVA expansion machinery and tested the accuracy and efficiency of the code by applying it to some classical test functions.

We defined an error functional of the solution to the BGK+PML model and applied the ANOVA expansion to this functional to calculate the Total Sensitivity Indices (TSIs) of the parameters. The bounds established through the stability analysis were used to choose carefully the parameter values for the simulations required to calculate the TSIs.

The sensitivity analysis allowed us to focus our attention on the most important parameters. The results showed that the most significant parameters in the BGK+PML model are the PML exponent β and the PML thickness L. We demonstrated that the TSIs are basically independent of the choice of the functional.

We used minimization techniques to choose the parameter values in an optimal

way, so as to minimize the error functional. We found that there is a region in which the error functional is in the order of 10^{-7}. Looking at the boundary of that region, we also observed that when β increases, L can decrease and still the error functional remains zero.

We showed that the form of the initial density distribution affects the response of the PML. In particular, we found that the magnitude of the error functional, and therefore the optimal PML thickness L, depends on the sharpness of the peak in the initial density distribution. We concluded that when the peak in the initial density distribution is sharper, the PML thickness L to attain the same level of accuracy is smaller. We note that we only changed the sharpness of the peak in the initial density distribution, but not its mathematical expression. It is reasonable to expect that by assuming another form of the initial density distribution, e.g. using a time-dependent source, one may obtain different results. In general, we emphasize that all the results obtained by applying the ANOVA analysis to the BGK+PML model depend on the problem configuration. We expect that by changing the problem settings, notably the initial conditions and/or the boundary conditions, one may obtain different results. These aspects deserve to be further investigated.

We finally recall that nowadays there is a well established theory for the development and the analysis of PMLs for linear problems only. Since the Navier-Stokes equations (NSE) have a nonlinear nature, one cannot directly apply that theory to develop a PML for them. Yet for low Mach numbers and weakly compressible flows one could couple the NSE and the BGK model, by solving the NSE in the physical domain and the BGK model in the PML domain. The coupling of the NSE and the BGK equations is an open research direction which is left for future investigation.

Appendices

Appendix A

Symmetrizers and well-posedness

In this appendix we will discuss how a general system of m first-order nonlinear PDEs can be symmetrized through left-multiplication with the Hessian matrix of an additional conserved variable. We will work out an additional conservation law for the BGK model and show that the BGK model is well-posed.

A.1 Additional conservation law

Consider a system of m first-order nonlinear conservation equations for m unknowns in the matrix form

$$\frac{\partial \boldsymbol{u}}{\partial t} + \sum_{k=1}^{d} \frac{\partial \boldsymbol{f}^k}{\partial x_k} = \mathbf{0}, \tag{A.1}$$

where $\boldsymbol{u}$ is the vector of conserved variables, $\boldsymbol{f}^k$ are the flux vectors, which are functions of $\boldsymbol{u}$, and d is the number of space dimensions. The system (A.1) can be rewritten as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \sum_{k=1}^{d} \boldsymbol{\nabla}_{\boldsymbol{u}} \boldsymbol{f}^k \frac{\partial \boldsymbol{u}}{\partial x_k} = \mathbf{0}, \tag{A.2}$$

where $\boldsymbol{\nabla}_{\boldsymbol{u}} \boldsymbol{f}^k$ denotes the gradient of the flux vector $\boldsymbol{f}^k$ with respect to the conserved variables $\boldsymbol{u}$, also known as Jacobian matrix of $\boldsymbol{f}^k$.

We consider the additional scalar conservation equation

$$\frac{\partial U}{\partial t} + \sum_{k=1}^{d} \frac{\partial F^k}{\partial x_k} = 0,$$

usually called *balance of entropy*. This name is due to the fact that usually the additional conserved variable coincides with the specific energy of the system, or, more often, with the entropy density.

The additional conservation equation and the new conserved variable U play a key role in the following remarkable result [9].

Theorem A.1 (Friedrichs and Lax)**.** *If a system of m conservation laws implies an additional conservation law such that the new conserved quantity U is a*

convex function of the original quantities $\boldsymbol{u}$, then it can be symmetrized by left-multiplication with the Hessian matrix of U, and the initial value problem (or Cauchy Problem) is well-posed.

Boillat [4] and later Larecki [20] showed that the converse of this result is also true.

We observe that this theorem can be interpreted as an *equivalence result between the existence of the entropy balance law and the symmetrizability of the system by Hessian matrices*. In other words, the symmetrizability condition for (A.1) corresponds to the condition that the system implies the balance of entropy.

Moreover, if the system (A.2) is already symmetric, namely if

$$\boldsymbol{\nabla}_{\boldsymbol{u}} \boldsymbol{f}^k = \left(\boldsymbol{\nabla}_{\boldsymbol{u}} \boldsymbol{f}^k\right)^T, \qquad k = 1, \dots, d,$$

then we can derive from it the new conservation law

$$\frac{\partial U}{\partial t} + \sum_{k=1}^{d} \frac{\partial F^k}{\partial x_k} = 0,$$

with

$$U = \frac{1}{2} \sum_{i=1}^{m} (u_i)^2, \qquad F^k = \boldsymbol{u} \cdot \boldsymbol{f}^k - g^k, \qquad k = 1, \dots, d,$$

where g^k satisfies

$$\frac{\partial g^k}{\partial u_i} = f_i^k, \qquad i = 1, \dots, m, \qquad k = 1, \dots, d.$$

A.1.1 Additional conservation law of the BGK model

Now we turn our attention to the homogeneous form of the BGK model, namely

$$\frac{\partial \boldsymbol{a}}{\partial t} + A_1 \frac{\partial \boldsymbol{a}}{\partial x_1} + A_2 \frac{\partial \boldsymbol{a}}{\partial x_2} = \boldsymbol{0}. \tag{A.3}$$

It should be clear that in our BGK model the matrices A_1 and A_2 are the Jacobian of the fluxes. Noting that (A.3) is indeed symmetric since A_1 and A_2 are both symmetric, one is tempted to apply Theorem A.1 to prove the well-posedness of the Cauchy problem for the BGK model. This will provide us an additional conserved variable that can be regarded as an entropy, and it will show how this symmetrization technique would work in the more general case of a non-symmetric system.

We first write the additional conserved variable as

$$U(\boldsymbol{a}) = \frac{1}{2} \sum_{i=0}^{5} (a_i)^2 = \frac{1}{2} \left(a_0^2 + a_1^2 + a_2^2 + a_3^2 + a_4^2 + a_5^2\right),$$

which is clearly convex. By Theorem A.1, we can already conclude that the Cauchy problem for (A.3) is well-posed.

Now, just for the sake of curiosity, we work out the additional conservation law. We calculate the fluxes for the BGK model, which turn out to be

$$\boldsymbol{f}^1 = \begin{bmatrix} a_1 \\ a_0 + \sqrt{2}\, a_4 \\ a_3 \\ a_2 \\ \sqrt{2}\, a_1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{f}^2 = \begin{bmatrix} a_2 \\ a_3 \\ a_0 + \sqrt{2}\, a_5 \\ a_1 \\ 0 \\ \sqrt{2}\, a_2 \end{bmatrix},$$

as can be easily checked by computing the Jacobian matrices $A_1 = \boldsymbol{\nabla}_{\boldsymbol{a}} \boldsymbol{f}^1$ and $A_2 = \boldsymbol{\nabla}_{\boldsymbol{a}} \boldsymbol{f}^2$. Here we simply set to zero all the integration constants.

To derive g^1, we exploit the equalities

$$\frac{\partial g^1}{\partial a_i} = f_i^1, \qquad i = 0, \ldots, 5,$$

which imply

$$g^1 = \left(a_0 + \sqrt{2}\, a_4\right) a_1 + a_2\, a_3.$$

Following the same procedure, one obtains for g^2

$$g^2 = \left(a_0 + \sqrt{2}\, a_5\right) a_2 + a_1\, a_3.$$

With g^1 and g^2 we can calculate the fluxes for the additional conserved variable

$$F^1 = \boldsymbol{a} \cdot \boldsymbol{f}^1 - g^1 = a_0\, a_1 + a_2\, a_3 + \sqrt{2}\, a_1\, a_4,$$

$$F^2 = \boldsymbol{a} \cdot \boldsymbol{f}^2 - g^2 = a_0\, a_2 + a_1\, a_3 + \sqrt{2}\, a_3\, a_5.$$

Finally, the additional conservation law is

$$\frac{\partial U}{\partial t} + \frac{\partial F^1}{\partial x_2} + \frac{\partial F^2}{\partial x_2} = 0,$$

with U, F^1 and F^2 having the expressions we just worked out.

A.2 Symmetrization

In the general case, the matrix H that makes the system (A.1) symmetric is given by the Hessian of $U(\boldsymbol{a})$. The BGK model (A.3) is already symmetric, so it must be the case that no symmetrizer is needed. Indeed, if we compute the Hessian of $U(\boldsymbol{a})$, we recover the identity matrix

$$H = \operatorname{Hess} U(\boldsymbol{a}) = I_{6\times 6}.$$

In the general case in which the original system (A.1) is not symmetric, and the number of dimensions d is equal to 2, one proceeds further by defining the matrix $A(\boldsymbol{n})$ as

$$A(\boldsymbol{n}) = n_1\, HA_1 + n_2\, HA_2,$$

where $\boldsymbol{n}$ is a 2D unit vector whose components are n_1, n_2, namely

$$\boldsymbol{n} = (n_1, n_2)\,, \qquad n_1^2 + n_2^2 = 1.$$

One can see $\boldsymbol{n}$ as the direction of wave propagation. We recall that in two dimensions a wave has infinitely-many directions of propagation[1]. One can then perform the spectral decomposition of $A(\boldsymbol{n})$ as

$$A(\boldsymbol{n}) = T\Lambda T^{-1},$$

where $T \equiv T(\boldsymbol{n})$ is a matrix composed of the eigenvectors of $A(\boldsymbol{n})$ and $\Lambda \equiv \Lambda(\boldsymbol{n})$ is the diagonal matrix constructed from the eigenvalues of $A(\boldsymbol{n})$.

The *characteristic variables* are defined as

$$\boldsymbol{z}(\boldsymbol{n}) = T^{-1}\boldsymbol{u},$$

where $\boldsymbol{u}$ contains the original conserved variables and T^{-1} is the inverse of the transformation matrix T.

In the case of the BGK model, since $H = I$, one defines $A(\boldsymbol{n})$ as

$$A(\boldsymbol{n}) = n_1\, A_1 + n_2\, A_2,$$

which coincides, as the reader can verify, with the matrix given in equation (1.7). Basically by following this procedure we go back to the analysis that we have carried out in Section 1.3 according to [7]. The eigenvalues of $A(\boldsymbol{n})$ are

$$0, \quad 0, \quad -\sqrt{n_1^2 + n_2^2}, \quad \sqrt{n_1^2 + n_2^2}, \quad -\sqrt{3}\sqrt{n_1^2 + n_2^2}, \quad \sqrt{3}\sqrt{n_1^2 + n_2^2}.$$

as in (1.8). We further observe that these eigenvalues are independent of $\boldsymbol{n}$ since by definition we chose $n_1^2 + n_2^2 = 1$. In other words, the eigenvalues of $A(\boldsymbol{n})$ simply are

$$0, \quad 0, \quad -1, \quad 1, \quad -\sqrt{3}, \quad \sqrt{3},$$

yet we emphasize that the characteristic variables $\boldsymbol{z}$ depend on the direction $\boldsymbol{n}$.

[1] We discussed these aspects in Section 1.6.

Appendix B

Notes on Hagstrom's paper, 2003

When dealing with numerical simulations of wave-dominated problems it is not possible of course to use an infinitely large domain to let the waves propagate freely, but it is necessary to truncate the computational boundary at some reasonable limits. In such cases, the introduction of absorbing boundary conditions allows to reduce the computational effort. During the last two decades, there has been a revival in absorbing layer techniques, mainly thanks to the introduction of perfectly matched layers by Bérenger [3]. As we have also mentioned in the main text, an improved approach to developing PMLs has later been proposed by Hagstrom [17].

The purpose of this appendix is to explain how to develop and analyse perfectly matched layers for general systems of constant coefficient hyperbolic PDEs. This presentation will mainly be based on the paper by Hagstrom [17].

We consider a two-dimensional problem, with an absorbing layer in the x-direction, having thickness L, as shown in Figure B.1.

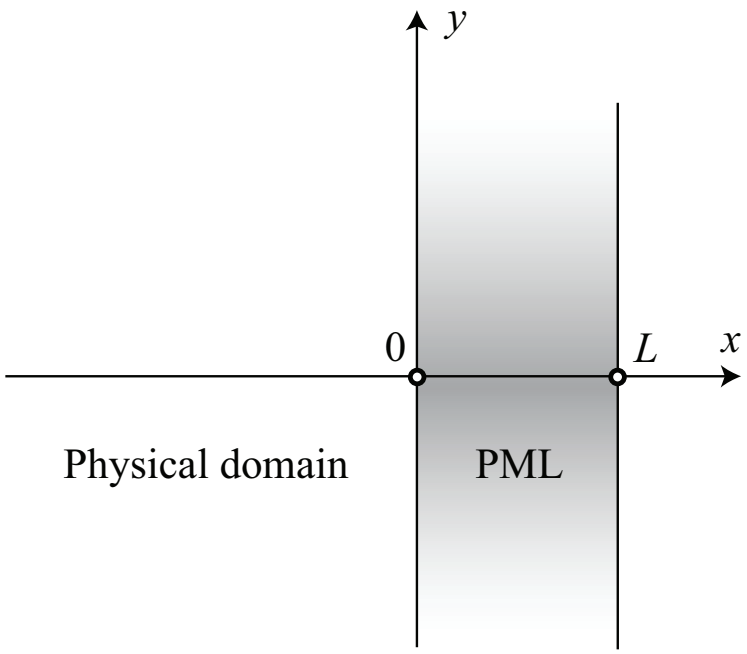


Figure B.1: Problem setting considered by Hagstrom.

A first order, constant coefficient hyperbolic system governs the solution outside the layer:

$$\frac{\partial \boldsymbol{u}}{\partial t} + A\frac{\partial \boldsymbol{u}}{\partial x} + B\frac{\partial \boldsymbol{u}}{\partial y} + C\boldsymbol{u} = \boldsymbol{0}. \tag{B.1}$$

It is intended that $\boldsymbol{u} \equiv \boldsymbol{u}(x, y, t)$. If we perform the Laplace transform of (B.1) in the time variable t we get:

$$s\hat{\boldsymbol{u}} + A\frac{\partial}{\partial x}\hat{\boldsymbol{u}} + B\frac{\partial}{\partial y}\hat{\boldsymbol{u}} + C\hat{\boldsymbol{u}} = \boldsymbol{0}, \tag{B.2}$$

where $s \in \mathbb{C}$ is the (complex) Laplace variable and $\hat{\boldsymbol{u}} \equiv \hat{\boldsymbol{u}}(x, y, s)$ denotes the Laplace transform of the sought solution $\boldsymbol{u}$. We have exploited the so-called derivative property of the Laplace transform, namely:

$$\mathcal{L}[f'(t)]_{(s)} = sF(s) - f(0),$$

with $F(s) = \mathcal{L}[f(t)]_{(s)}$.

Now we make the assumption that the Laplace transform of the solution can be written as a separable variable solution:

$$\hat{\boldsymbol{u}} = e^{\lambda x}\boldsymbol{\phi}, \tag{B.3}$$

with $\boldsymbol{\phi}$ being functions of s and y, but not of x, namely $\boldsymbol{\phi} \equiv \boldsymbol{\phi}(s, y)$. In general, the coefficient λ can also be in $\mathbb{C}$.

If we substitute (B.3) into (B.2) we get:

$$se^{\lambda x}I\boldsymbol{\phi} + A\frac{\partial}{\partial x}\left(e^{\lambda x}\boldsymbol{\phi}\right) + B\frac{\partial}{\partial y}\left(e^{\lambda x}\boldsymbol{\phi}\right) + Ce^{\lambda x}\boldsymbol{\phi} = \boldsymbol{0},$$

$$se^{\lambda x}I\boldsymbol{\phi} + A\,\lambda e^{\lambda x}\boldsymbol{\phi} + B\,e^{\lambda x}\frac{\partial}{\partial y}\boldsymbol{\phi} + Ce^{\lambda x}\boldsymbol{\phi} = \boldsymbol{0},$$

$$\left(sI + \lambda A + B\frac{\partial}{\partial y} + C\right)\boldsymbol{\phi} = \boldsymbol{0}. \tag{B.4}$$

This is the *modal equation outside the layer*. The $\boldsymbol{\phi}$ are the eigenfunctions, or *eigenmodes*. It is advisable to rewrite (B.4) as:

$$\left(sI + B\frac{\partial}{\partial y} + C\right)\boldsymbol{\phi} = -\lambda A\,\boldsymbol{\phi}, \tag{B.5}$$

so that we can readily recognize this as a *generalized eigenvalue problem with eigenfunctions* $\boldsymbol{\phi}$.

The λ can be easily worked out as:

$$\lambda = \frac{-\left(sI + \mathrm{i}k_2 B + C\right)\boldsymbol{\phi}}{A\boldsymbol{\phi}}.$$

We take the real part of λ

$$\mathrm{Re}(\lambda) = -\frac{(\mathrm{Re}(s)I + C)\,\boldsymbol{\phi}}{A\boldsymbol{\phi}}$$

and left-multiply by the complex conjugate of $\boldsymbol{\phi}$, obtaining

$$\mathrm{Re}(\lambda) = -\frac{\mathrm{Re}(s) + |C|_2}{|A|_2}, \tag{B.6}$$

where $|C|_2 = \boldsymbol{\phi}^* C\boldsymbol{\phi}$, $|A|_2 = \boldsymbol{\phi}^* A\boldsymbol{\phi}$ and we have assumed that $\boldsymbol{\phi}$ is normalized, i.e. $\boldsymbol{\phi}^*\boldsymbol{\phi} = 1$.

Now let us look at the numerator of (B.6). We note that if $\mathrm{Re}(s)$ is sufficiently large, i.e. if $\mathrm{Re}(s) > |C|_2$, then there are no purely imaginary eigenvalues λ. In this case, we may assume that the eigenvalues fall into two sets, one containing eigenvalues with negative real part and the other containing eigenvalues with positive real part.

Instead, if $C = 0$, then in the limit $\mathrm{Re}(s) \to 0$ one also has $\mathrm{Re}(\lambda) \to 0$, but we really want to avoid purely imaginary eigenvalues. We have to come up with a layer model such that $\mathrm{Re}(\lambda)$ does not tend to zero when $\mathrm{Re}(s) \to 0$ and so that there are no reflections at the interface between the layer and the physical domain. To avoid reflections at the interface is necessary to build the governing equations of the layer in such a way that, after Laplace-transforming them, their eigenfunctions are the same as outside the layer.

Hagstrom proposed the following governing equations for the PML:

$$\frac{\partial \boldsymbol{u}}{\partial t} + A\left(\frac{\partial \boldsymbol{u}}{\partial x} + \boldsymbol{v} + \boldsymbol{w}\right) + B\,\frac{\partial \boldsymbol{u}}{\partial y} + C\boldsymbol{u} = \boldsymbol{0}, \tag{B.7}$$

$$R\boldsymbol{w} + \sigma\boldsymbol{w} + \sigma\left(\frac{\partial \boldsymbol{u}}{\partial x} + \boldsymbol{v}\right) = \boldsymbol{0}, \tag{B.8}$$

$$M\boldsymbol{v} = \sigma N\boldsymbol{u}, \tag{B.9}$$

where $\boldsymbol{w}, \boldsymbol{v}$ are auxiliary variables, M, N are numbers, $\sigma \geq 0$ is the absorption parameter and R is a scalar differential operator:

$$R := \frac{\partial}{\partial t} + \beta\frac{\partial}{\partial y} + \alpha.$$

Note that the only difference between (B.1) and (B.7) are the auxiliary variables trailing the term $\partial\boldsymbol{u}/\partial x$.

Hagstrom suggests that the modal solution inside the layer is given by the Ansatz:

$$\hat{\boldsymbol{u}} = e^{\lambda x + \left(\lambda\hat{R}^{-1} - \hat{M}^{-1}\hat{N}\right)\int_0^x \sigma(z)\,\mathrm{d}z}\boldsymbol{\phi}, \tag{B.10}$$

and then he claims that "when we substitute (B.10) into the new system we want the eigenvalue problem for $\boldsymbol{\phi}$ in (B.4) to be the result". To make this clearer: when we substitute the Ansatz (B.10) into the *Laplace transform* of the PML system (B.7), (B.8), (B.9) *and set* $\sigma = 0$ we should get the eigenvalue problem (B.4) for $\boldsymbol{\phi}$ outside

the layer. This is exactly what we are looking for: we want the eigenfunctions for the eigenvalue problem *inside* the layer to be the same that satisfy the eigenvalue problem *outside* the layer.

In the following, we are going to check that this is actually the case, since Hagstrom states it, but leaves the calculations to the reader. The first thing to do is to make the auxiliary variables disappear from our equations. From (B.9) we work out $\boldsymbol{v}$:

$$\boldsymbol{v} = \sigma M^{-1} N \boldsymbol{u}, \tag{B.11}$$

then $\boldsymbol{w}$ from (B.8):

$$\begin{aligned}
(R+\sigma)\,\boldsymbol{w} + \sigma\left(\frac{\partial \boldsymbol{u}}{\partial x} + \boldsymbol{v}\right) &= \mathbf{0}, \\
(R+\sigma)\,\boldsymbol{w} + \sigma\left(\frac{\partial \boldsymbol{u}}{\partial x} + \sigma M^{-1} N \boldsymbol{u}\right) &= \mathbf{0}, \\
\boldsymbol{w} = -\sigma\,(R+\sigma)^{-1}\left(\frac{\partial}{\partial x} + \sigma M^{-1} N\right)\boldsymbol{u}. &
\end{aligned} \tag{B.12}$$

Now we insert (B.11) and (B.12) into (B.7):

$$\frac{\partial \boldsymbol{u}}{\partial t} + A\left(\frac{\partial \boldsymbol{u}}{\partial x} + \sigma M^{-1} N \boldsymbol{u} - \sigma\,(R+\sigma)^{-1}\left(\frac{\partial}{\partial x} + \sigma M^{-1} N\right)\boldsymbol{u}\right) + B\,\frac{\partial \boldsymbol{u}}{\partial y} + C\boldsymbol{u} = \mathbf{0},$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + A\left(\left(\frac{\partial}{\partial x} + \sigma M^{-1} N\right)\boldsymbol{u} - \sigma\,(R+\sigma)^{-1}\left(\frac{\partial}{\partial x} + \sigma M^{-1} N\right)\boldsymbol{u}\right) + B\,\frac{\partial \boldsymbol{u}}{\partial y} + C\boldsymbol{u} = \mathbf{0},$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + A\left(1 - \sigma\,(R+\sigma)^{-1}\right)\left(\frac{\partial}{\partial x} + \sigma M^{-1} N\right)\boldsymbol{u} + B\,\frac{\partial \boldsymbol{u}}{\partial y} + C\boldsymbol{u} = \mathbf{0}. \tag{B.13}$$

We point out that this is *a single vector equation for the PML model* obtained from the three coupled equations (B.7), (B.8) and (B.9). After Laplace-transforming in the variable t, the last equation turns into:

$$\left(sI + A\left(1 - \sigma(\hat{R}+\sigma)^{-1}\right)\left(\frac{\partial}{\partial x} + \sigma \hat{M}^{-1}\hat{N}\right) + B\,\frac{\partial}{\partial y} + C\right)\hat{\boldsymbol{u}} = \mathbf{0}.$$

We now insert the Ansatz for the modal solution inside the layer into the previous equation. For simplicity, we can assume that σ is constant. After some simple calculations, we get:

$$\left(sI + \lambda A\left(1 + \hat{R}^{-1}\sigma\right)\left(1 - \sigma(\hat{R}+\sigma)^{-1}\right) + B\,\frac{\partial}{\partial y} + C\right)\boldsymbol{\phi} = \mathbf{0}. \tag{B.14}$$

This is the *modal equation inside the layer*. It is now a simple matter to see that outside the layer, namely when $\sigma = 0$, the last equation reduces to (B.4). In other words, the eigenfunctions $\boldsymbol{\phi}$ remain the same, no matter if we are looking at the solution inside or outside the absorbing layer. This is exactly what we wanted, and it concludes the proof of the claim.

Now we want to analyse the decaying of the solution in time. We will take advantage of the fact that, when dealing with a constant coefficient case, we can also perform a Fourier transform in the spatial variables. Usually this is done in the variable(s) transversal to the direction of development of the layer. The reason for taking the Fourier Transform is that we get a nice polynomial in the Fourier variable. In other words, the integral transformations allow to recast a differential problem into an algebraic problem. If the coefficients are not constant, then this technique is no more useful.

Outside the layer we have

$$\frac{\partial \boldsymbol{u}}{\partial t} + A\frac{\partial \boldsymbol{u}}{\partial x} + B\frac{\partial \boldsymbol{u}}{\partial y} + C\boldsymbol{u} = \boldsymbol{0},$$

Fourier-transforming in all the spatial variables

$$\frac{\partial \tilde{\boldsymbol{u}}}{\partial t} + \mathrm{i}k_1 A\tilde{\boldsymbol{u}} + \mathrm{i}k_2 B\tilde{\boldsymbol{u}} + C\tilde{\boldsymbol{u}} = \boldsymbol{0},$$

we get the system of ordinary differential equations:

$$\frac{\partial \tilde{\boldsymbol{u}}}{\partial t} = -\left(\mathrm{i}k_1 A + \mathrm{i}k_2 B + C\right)\tilde{\boldsymbol{u}},$$

whose solution is given by:

$$\tilde{\boldsymbol{u}} = c\,\exp\left[-\left(\mathrm{i}k_1 A + \mathrm{i}k_2 B + C\right)t\right]. \tag{B.15}$$

Assuming that A and B contain only real coefficients, we have that the terms $\mathrm{i}k_1 A$ and $\mathrm{i}k_2 B$ are pure imaginary, so we do not care about them, since the complex exponential is only a linear combination of sine and cosine waves, and thus shows the oscillatory wave-like behaviour. What we really wish is that $\mathrm{Re}\left[-\left(\mathrm{i}k_1 A + \mathrm{i}k_2 B + C\right)\right] < 0$, which means that we have to ensure that $\mathrm{Re}(C) > 0$. In other words, if all the coefficients in C are strictly positive, then we actually have a decaying wave with respect to time, a behaviour which agrees with the physics of the problem.

Now consider the single equation (B.13) that governs the solution inside the layer

$$\frac{\partial \boldsymbol{u}}{\partial t} + A\left(1 - \sigma\left(R+\sigma\right)^{-1}\right)\left(\frac{\partial}{\partial x} + \sigma M^{-1}N\right)\boldsymbol{u} + B\frac{\partial \boldsymbol{u}}{\partial y} + C\boldsymbol{u} = \boldsymbol{0},$$

that we have worked out before. If we take the Fourier transform in all the spatial coordinates:

$$\frac{\partial \tilde{\boldsymbol{u}}}{\partial t} + A\left(1 - \sigma\widetilde{\left(R+\sigma\right)^{-1}}\right)\left(\mathrm{i}k_1 + \sigma M^{-1}N\right)\tilde{\boldsymbol{u}} + \mathrm{i}k_2 B\tilde{\boldsymbol{u}} + C\tilde{\boldsymbol{u}} = \boldsymbol{0},$$

$$\frac{\partial \tilde{\boldsymbol{u}}}{\partial t} = -\left[A\left(1 - \sigma\widetilde{\left(R+\sigma\right)^{-1}}\right)\left(\mathrm{i}k_1 + \sigma M^{-1}N\right) + \mathrm{i}k_2 B + C\right]\tilde{\boldsymbol{u}}.$$

The solution of this equation is similar to (B.15). We emphasize though that $(R+\sigma_1)^{-1}$ is the inverse of a scalar differential operator, and in general to work out this kind of inverse is not a trivial task. In general, one has to resort to the theory of pseudodifferential operators.

At any rate, since we are mapping the differential operators into the Laplace-Fourier space, we note that *the transform of the inverse of a differential operator is equal to the reciprocal of the transform of the differential operator*. Symbolically, if we denote by $P(\partial/\partial x)$ a differential operator, the following result holds:

$$\mathcal{F}\left[P^{-1}(\partial/\partial x)\right] = \frac{1}{\mathcal{F}\left[P(\partial/\partial x)\right]},$$

where $\mathcal{F}$ denotes the Fourier transform and $P^{-1}(\partial/\partial x)$ is the inverse of the differential operator $P(\partial/\partial x)$. It can be showed that this same property holds also for the Laplace transform.

Specializing it to our case, it should by now be clear that

$$\mathcal{LF}\left[(R+\sigma_1)^{-1}\right] = \frac{1}{\mathcal{LF}\left[R+\sigma_1\right]},$$

where $\mathcal{LF}$ denotes the Laplace-Fourier transform. Recalling the definition of R, we get the result:

$$\mathcal{LF}\left[(R+\sigma_1)^{-1}\right] = \frac{1}{\mathcal{LF}\left[\dfrac{\partial}{\partial t} + \alpha_1 \dfrac{\partial}{\partial x_2} + \alpha_0 + \sigma_1\right]} = \frac{1}{s + \mathrm{i}\alpha_1 k_2 + \alpha_0 + \sigma_1}.$$

After all this discussion about integral transforms, the following questions may naturally arise: *When do we take the Fourier transform in only one spatial variable, and when do we take it in both spatial variables?* And also, *When do we take the Laplace Transform and when we do not?* The answers are as follows:

- When we want to come up with an Ansatz for the modal solution inside the layer, then we take the Laplace transform in t and the Fourier transform in the spatial variable which is transversal with respect to the direction of development of the layer.
- When we want to prove stability through bounds on the energy decay, then we perform the Fourier transform in all spatial variables (see Section 3.2). In this case, we can also do the Laplace transform, but it is not necessary and also makes things a bit more complicated, because in that case we also have to prove that the solution is well-behaved for any $\mathrm{Re}(s) > 0$.

We point out once more that the Laplace-Fourier transformation machinery allows turn a differential problem into an algebraic one. Notice that this is a general approach often used when dealing with wave propagation phenomena: we switch from the wave equation (which is a PDE) to the so-called Helmholtz equation (which is an ODE).

Appendix C

Modal analysis in Laplace-Fourier space

In this appendix we will apply the theory discussed in Appendix B to the case of the BGK model.

The governing equation of the BGK variables inside the layer is:

$$\frac{\partial \boldsymbol{a}}{\partial t} + A_1 \left(\frac{\partial \boldsymbol{a}}{\partial x_1} + \sigma_1 \left(\lambda_0 \boldsymbol{a} + \boldsymbol{\omega} \right) \right) + A_2 \left(\frac{\partial \boldsymbol{a}}{\partial x_2} + \sigma_2 \left(\tilde{\lambda}_0 \boldsymbol{a} + \boldsymbol{\theta} \right) \right) = S(\boldsymbol{a}). \tag{C.1}$$

First we want to make the auxiliary variables disappear from this equation. We therefore turn our attention for a moment to the equations governing the evolution of the auxiliary variables. The auxiliary variable $\boldsymbol{\omega}$ is evolved according to:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \alpha_1 \frac{\partial \boldsymbol{\omega}}{\partial x_2} + (\alpha_0 + \sigma_1)\boldsymbol{\omega} + \frac{\partial \boldsymbol{a}}{\partial x_1} + \lambda_0(\alpha_0 + \sigma_1)\boldsymbol{a} - \lambda_1 \frac{\partial \boldsymbol{a}}{\partial x_2} = \boldsymbol{0}.$$

We define the following first order scalar differential operator:

$$R := \frac{\partial}{\partial t} + \alpha_1 \frac{\partial}{\partial x_2} + \alpha_0,$$

so that the previous equation becomes:

$$R\boldsymbol{\omega} + \sigma_1 \boldsymbol{\omega} + \frac{\partial \boldsymbol{a}}{\partial x_1} + \lambda_0(\alpha_0 + \sigma_1)\boldsymbol{a} - \lambda_1 \frac{\partial \boldsymbol{a}}{\partial x_2} = \boldsymbol{0}.$$

Similarly, the auxiliary variable $\boldsymbol{\theta}$ is evolved according to:

$$\frac{\partial \boldsymbol{\theta}}{\partial t} + \tilde{\alpha}_1 \frac{\partial \boldsymbol{\theta}}{\partial x_1} + (\tilde{\alpha}_0 + \sigma_2)\boldsymbol{\theta} + \frac{\partial \boldsymbol{a}}{\partial x_2} + \tilde{\lambda}_0(\tilde{\alpha}_0 + \sigma_2)\boldsymbol{a} - \tilde{\lambda}_1 \frac{\partial \boldsymbol{a}}{\partial x_1} = \boldsymbol{0}.$$

Also we define a first order scalar differential operator:

$$M := \frac{\partial}{\partial t} + \tilde{\alpha}_1 \frac{\partial}{\partial x_1} + \tilde{\alpha}_0,$$

so that:

$$M\boldsymbol{\theta} + \sigma_2 \boldsymbol{\theta} + \frac{\partial \boldsymbol{a}}{\partial x_2} + \tilde{\lambda}_0(\tilde{\alpha}_0 + \sigma_2)\boldsymbol{a} - \tilde{\lambda}_1 \frac{\partial \boldsymbol{a}}{\partial x_1} = \boldsymbol{0}.$$

$$(R+\sigma_1)\boldsymbol{\omega} = -\frac{\partial \boldsymbol{a}}{\partial x_1} - \lambda_0(\alpha_0+\sigma_1)\boldsymbol{a} + \lambda_1\frac{\partial \boldsymbol{a}}{\partial x_2},$$

$$\boldsymbol{\omega} = (R+\sigma_1)^{-1}\left(-\frac{\partial \boldsymbol{a}}{\partial x_1} - \lambda_0(\alpha_0+\sigma_1)\boldsymbol{a} + \lambda_1\frac{\partial \boldsymbol{a}}{\partial x_2}\right),$$

where the $^{-1}$ denotes the inverse operator. Analogously:

$$(M+\sigma_2)\boldsymbol{\theta} = -\frac{\partial \boldsymbol{a}}{\partial x_2} - \tilde{\lambda}_0(\tilde{\alpha}_0+\sigma_2)\boldsymbol{a} + \tilde{\lambda}_1\frac{\partial \boldsymbol{a}}{\partial x_1},$$

$$\boldsymbol{\theta} = (M+\sigma_2)^{-1}\left(-\frac{\partial \boldsymbol{a}}{\partial x_2} - \tilde{\lambda}_0(\tilde{\alpha}_0+\sigma_2)\boldsymbol{a} + \tilde{\lambda}_1\frac{\partial \boldsymbol{a}}{\partial x_1}\right).$$

We insert these expressions into (C.1):

$$\begin{aligned}&\frac{\partial \boldsymbol{a}}{\partial t} + A_1\left(\frac{\partial \boldsymbol{a}}{\partial x_1} + \sigma_1\left(\lambda_0\boldsymbol{a} + (R+\sigma_1)^{-1}\left(-\frac{\partial \boldsymbol{a}}{\partial x_1} - \lambda_0(\alpha_0+\sigma_1)\boldsymbol{a} + \lambda_1\frac{\partial \boldsymbol{a}}{\partial x_2}\right)\right)\right)\\ &\quad + A_2\left(\frac{\partial \boldsymbol{a}}{\partial x_2} + \sigma_2\left(\tilde{\lambda}_0\boldsymbol{a} + (M+\sigma_2)^{-1}\left(-\frac{\partial \boldsymbol{a}}{\partial x_2} - \tilde{\lambda}_0(\tilde{\alpha}_0+\sigma_2)\boldsymbol{a} + \tilde{\lambda}_1\frac{\partial \boldsymbol{a}}{\partial x_1}\right)\right)\right) = S(\boldsymbol{a}),\end{aligned}$$

and after going through a little bit of algebra, we get

$$\begin{aligned}&\frac{\partial \boldsymbol{a}}{\partial t} + A_1\left(\left(I-\sigma_1(R+\sigma_1)^{-1}\right)\left(\frac{\partial}{\partial x_1}+\sigma_1\lambda_0\right) + \sigma_1(R+\sigma_1)^{-1}\left(\lambda_1\frac{\partial}{\partial x_2} - \lambda_0\alpha_0\right)\right)\boldsymbol{a}\\ &\quad + A_2\left(\left(I-\sigma_2(M+\sigma_2)^{-1}\right)\left(\frac{\partial}{\partial x_2}+\sigma_2\tilde{\lambda}_0\right) + \sigma_2(M+\sigma_2)^{-1}\left(\tilde{\lambda}_1\frac{\partial}{\partial x_1} - \tilde{\lambda}_0\tilde{\alpha}_0\right)\right)\boldsymbol{a} = S(\boldsymbol{a}).\end{aligned}$$

If the layer develops in the x_1-direction only, namely if $\sigma_2 = 0$, then we have

$$\begin{aligned}&\frac{\partial \boldsymbol{a}}{\partial t} + A_1\left(\left(I-\sigma_1(R+\sigma_1)^{-1}\right)\left(\frac{\partial}{\partial x_1}+\sigma_1\lambda_0\right) + \sigma_1(R+\sigma_1)^{-1}\left(\lambda_1\frac{\partial}{\partial x_2} - \lambda_0\alpha_0\right)\right)\boldsymbol{a}\\ &\quad + A_2\frac{\partial \boldsymbol{a}}{\partial x_2} = S(\boldsymbol{a}).\end{aligned}$$

Now if we take the Laplace transform in t and the Fourier transform in x_2 (neglecting the term on the right-hand side) and introduce the symbol of R

$$\hat{r} = s + \mathrm{i}\alpha_1 k_2 + \alpha_0,$$

then the last equation becomes

$$\left(sI + A_1\left(\left(I - \frac{\sigma_1}{\hat{r}+\sigma_1}\right)\left(\frac{\partial}{\partial x_1}+\sigma_1\lambda_0\right) + \frac{\sigma_1}{\hat{r}+\sigma_1}\left(\lambda_1\mathrm{i}k_2 - \lambda_0\alpha_0\right)\right) + \mathrm{i}k_2A_2\right)\hat{\boldsymbol{a}} = \boldsymbol{0}. \tag{C.2}$$

Now this expression present a term that looks like the exponent in equation (22) in Gao et al. [11].

Appendix D

Routh-Hurwitz stability criterion

In control theory, the Routh-Hurwitz stability criterion is a test to check the stability of linear time-invariant control systems. In 1876, the English mathematician Edward Routh proposed a test to determine whether the roots of a characteristic polynomial of a linear system have negative real parts. A couple of decades later, in 1895, the German mathematician Adolf Hurwitz proposed to arrange the coefficients of the polynomial into a square matrix, called the Hurwitz matrix, and proved that the system is stable if and only if the sequence of determinants of its principal submatrices are all positive. It turns out that the two procedures are equivalent, which is why one talks about the Routh-Hurwitz stability criterion. The importance of this criterion lies in the fact that the roots λ of the characteristic equation of a linear system represent solutions of the type $e^{\lambda t}$. Therefore if all of these λ have negative real part, then the solutions are stable. *A polynomial satisfying the Routh-Hurwitz stability criterion is said to be Hurwitz-stable.*

The Routh test can be derived through the use of the *Euclidean algorithm* and *Sturm sequences*. These are exactly the same tools that lie behind Theorem 3.3, as the reader can verify in the bibliography, for instance by looking at the book of Marden [22]. In this appendix we show that the Routh-Hurwitz test and Theorem 3.3 develop on the same foundations.

We start off by stating the Routh-Hurwitz theorem.

Theorem D.1 (Routh-Hurwitz)**.** *Let $f(z)$ be a polynomial of degree n and*

$$f(\mathrm{i}y) = P_0 + \mathrm{i}P_1(y),$$

for a real y. Let p be the number of roots of the polynomial $f(z)$ with negative real part and q the number of roots with positive real part. Then, if there are no roots lying on the imaginary axis, the following result holds

$$p - q = w(+\infty) - w(-\infty),$$

where $w(x)$ is the number of variations of the generalized Sturm chain obtained from $P_0(y)$ and $P_1(y)$ by successive Euclidean divisions.

We point out that the fundamental theorem of algebra states that each polynomial of degree n has exactly n roots in the complex plane. If $f(z)$ has no roots on the

imaginary axis, then we can write $n = p + q$. We have that $f(z)$ is a Hurwitz-stable polynomial if and only if $p \equiv n$ (or, equivalently, if $q \equiv 0$). Using the Routh-Hurwitz theorem one can replace the condition on p and q by a condition on the *generalized Sturm chain*, which yields a condition on the coefficients of the polynomial.

Now we turn our attention to the aspects behind Theorem 3.3, used by Appelö et al. [2]. Theorem 3.3 comes from a special case of corollary (38.1b) in Marden [22]. In fact, Chapter IX in [22] deals with the problem of finding the exact or approximate number of zeros which lie in a prescribed region such as a half-plane, a sector or a circular region. The presentation in [22] covers how this problem can arise from physics and applied mathematics by an example taken from Routh. Marden shows the example of a system whose sufficient condition for stability is to have all the roots of its characteristic polynomial lying in the left half-plane (i.e., having negative signs). To determine the number of zeros of a polynomial in a given half-plane, the concept of *Cauchy index* is introduced. Here we limit ourselves to state the *Cauchy Index Theorem* as presented in [22], essentially in the same form given by Hurwitz.

Theorem D.2 (Cauchy Index Theorem)**.** *Let*

$$f(z) = a_0 + a_1 z + \cdots + a_{n-1} z^{n-1} + z^n = P_0(z) + \mathrm{i} P_1(z),$$

where $P_0(z)$ and $P_1(z)$ are real polynomials with $P_1(z) \neq 0$. As the point $z = x$ moves on the real axis from $-\infty$ to $+\infty$, let σ be the number of real zeros of $P_0(z)$ at which $\rho(x) = P_0(x)/P_1(x)$ changes from $-$ to $+$, and τ the number of real zeros of $P_0(z)$ at which $\rho(x)$ changes from $+$ to $-$. If $f(z)$ has no real zeros, p zeros in the upper half-plane and q zeros in the lower half-plane, then

$$p = \frac{1}{2}\left[n + (\tau - \sigma)\right], \qquad q = \frac{1}{2}\left[n - (\tau - \sigma)\right].$$

The Cauchy Index Theorem turns the problem of finding the number of zeros in the upper and lower half-planes into the problem of calculating the difference $(\tau - \sigma)$. In the case of real polynomials, this difference has been computed with the theory of residues by Hurwitz and with the use of Sturm chains by Routh.

Following the approach of Routh, one can construct the Sturm sequence of functions $P_0(x), P_1(x), P_2(x), \ldots, P_\mu(x)$ by applying to $P_0(x)$ and P_1 the Euclidean division algorithm in which the remainder is written with a negative sign. We define $w\{P_k(x)\} \equiv w\{P_0(x), P_1(x), P_2(x), \ldots, P_\mu(x)\}$ the number of variations of sign in the sequence $P_0(x), P_1(x), P_2(x), \ldots, P_\mu(x)$. One can show that

$$\tau - \sigma = w\{P_k(+\infty)\} - w\{P_k(-\infty)\}$$

and, with this result in mind, one can restate the Cauchy Index Theorem as follows.

Theorem D.3. *Let*

$$f(z) = a_0 + a_1 z + \cdots + a_{n-1} z^{n-1} + z^n = P_0(z) + \mathrm{i} P_1(z),$$

where $P_0(z)$ and $P_1(z)$ are real polynomials with $P_1(z) \neq 0$, be a polynomial which has no real zeros, p zeros in the upper half-plane and q zeros in the lower half-plane. Let $P_0(x), P_1(x), P_2(x), \ldots, P_\mu(x)$ be the Sturm sequence formed by applying to $P_0(x)/P_1(x)$ the negative-remainder, Euclidean division algorithm. Then

$$p = \frac{1}{2}\left[n + w\{P_k(+\infty)\} - w\{P_k(-\infty)\}\right],$$

$$q = \frac{1}{2}\left[n - w\{P_k(+\infty)\} + w\{P_k(-\infty)\}\right].$$

We finally have the following result, given as corollary (38,1b) in [22].

Theorem D.4. *If for $P_0(x)$ and $P_1(x)$ of Theorem D.3 there is a continued fraction expansion*

$$\frac{P_1(x)}{P_0(x)} = \cfrac{1}{c_1 x + d_1 - \cfrac{1}{c_2 x + d_2 - \cfrac{1}{c_3 x + d_3 - \cdots - \cfrac{1}{c_{n_r} x + d_{n_r}}}}}$$

where $c_j \neq 0$ for $j = 1, 2, \ldots, n$, then p is equal to the number of coefficients c_j having negative sign, while q is equal to the number of coefficients c_j having positive sign.

This result is due to Wall in the case of real polynomials and to Frank [8] in the case of complex polynomials. It is exactly Theorem 3.3 that we used to study the stability of the BGK+PML model in Section 3.3.

Bibliography

[1] T. Andres, *Sampling methods and sensitivity analysis for large parameter sets*, Journal of Statistical Computation and Simulation, 57 (1997), pp. 77–110.

[2] D. Appelö, T. Hagstrom, and G. Kreiss, *Perfectly Matched Layers for Hyperbolic Problems: General Formulation, Well-Posedness and Stability*, SIAM Journal on Applied Mathematics, 67 (2006), pp. 1–23.

[3] J.-P. Bérenger, *A Perfectly Matched Layer for the Absorption of Electromagnetic Waves*, Journal of Computational Physics, 114 (1994), pp. 185–200.

[4] G. Boillat, *Sur l'existence et la recherche d'équations de conservation supplémentaires pour les systèmes hyperboliques*, C.R. Acad. Sci. Paris, 278 A (1974), pp. 909–914.

[5] Y. Cao, Z. Chen, and M. Gunzburger, *ANOVA Expansions and Efficient Sampling Methods for Parameter Dependent Nonlinear PDEs*, International Journal of Numerical Analysis and Modeling, 6 (2009), pp. 256–273.

[6] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration: Second Edition*, Academic Press, 1984.

[7] L. Evans, *Partial Differential Equations*, Graduate studies in mathematics, American Mathematical Society, 2010.

[8] E. Frank, *On the zeros of polynomials with complex coefficients*, Bull. Amer. Math. Soc., 52 (1946), pp. 144–157.

[9] K. O. Friedrichs and P. Lax, *Systems of Conservation Equations with a Convex Extension*, Proceedings of the National Academy of Sciences USA, 68 (1971), pp. 1686–1688.

[10] Z. Gao and J. S. Hesthaven, *Efficient Solution of Ordinary Differential Equations with High-Dimensional Parametrized Uncertainty*, Communications in Computational Physics, 10 (2011), pp. 253–278.

[11] Z. Gao, J. S. Hesthaven, and T. Warburton, *Efficient Absorbing Layers for Weakly Compressible Flows*. 2011.

[12] A. Genz, *Testing multidimensional integration routines*, in Proc. Of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation, New York, NY, USA, 1984, Elsevier North-Holland, Inc., pp. 81–94.

[13] ——, *A package for testing multiple integration subroutines*, in Numerical Integration, P. Keast and G. Fairweather, eds., vol. 203 of NATO ASI Series, Springer Netherlands, 1987, pp. 337–340.

[14] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong Stability-Preserving High-Order Time Discretization Methods*, SIAM Review, 43 (2001), pp. 89–112.

[15] H. Grad, *On the kinetic theory of rarefied gases*, Communications on Pure and Applied Mathematics, 2 (1949), pp. 331–407.

[16] B. Gustafsson, H.-O. Kreiss, and J. Oliger, *Time Dependent Problems and Difference Methods*, John Wiley and Sons, 1995.

[17] T. Hagstrom, *A New Construction of Perfectly Matched Layers for Hyperbolic Systems with Applications to the Linearized Euler Equations*, Mathematical and Numerical Aspects of Wave Propagation WAVES 2003, (2003), pp. 125–129.

[18] D. C. Hernquist, *Smoothly Symmetrizable Hyperbolic Systems of Partial Differential Equations*, Mathematica Scandinavica, 61 (1987), pp. 262–275.

[19] A. R. Krommer and C. Ueberhuber, *Numerical Integration on Advanced Computer Systems*, Academic Press, 1994.

[20] W. Larecki, *Symmetrization of systems of conservation equations and the converse to the condition of Friedrichs and Lax*, Arch. Mech., 49 (1997), pp. 865–876.

[21] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, SIAM, 2007.

[22] M. Marden, *Geometry of Polynomials*, American Mathematical Society, 1966.

[23] L. Rezzolla, *Numerical Methods for the Solution of Hyperbolic Partial Differential Equations*, (2005).

[24] A. Saltelli, K. Chan, and E. Scott, *Sensitivity Analysis*, Wiley, Chichester, 2000.

[25] A. H. Stroud, *Remarks on the Disposition of Points in Numerical Integration Formulas*, Mathematical Tables and Other Aids to Computation, 11 (1957), pp. 257–261.

[26] D. Xiu, *Numerical integration formulas of degree two*, Applied Numerical Mathematics, 58 (2008), pp. 1515–1520.